

全国计算机技术与软件专业技术资格（水平）考试指定用书

网络工程师教程

（第三版）（修订版）

雷震甲　主编

全国计算机专业技术资格考试办公室组编

清华大学出版社

全国计算机技术与软件专业技术资格（水平）考试指定用书

网络工程师教程

（第三版）（修订版）

雷震甲 严体华 吴晓葵 编著
全国计算机专业技术资格考试办公室 组编

清华大学出版社
北 京

内 容 简 介

本书是全国计算机技术与软件专业技术资格考试指定用书。本教材根据第三版的内容，并根据考试的重点内容做了修订，书中主要内容包括：数据通信、广域通信网、局域网、城域网、因特网、网络安全、网络操作系统与应用服务器配置、组网技术、网络管理和网络规划和设计。

本书是参加本考试的必备教材，也可作为网络工程从业人员学习网络技术的教材或日常工作的参考用书。

图书在版编目（CIP）数据

网络工程师教程（第三版）（修订版）/雷震甲主编. —北京：清华大学出版社，2011.9
（全国计算机技术与软件专业技术资格（水平）考试指定用书）
ISBN 978-7-302-26658-7

Ⅰ. ①网…　Ⅱ. ①雷…　Ⅲ. ①计算机网络-工程技术人员-资格考试-教材　Ⅳ. ①TP393

中国版本图书馆 CIP 数据核字（2011）第 179880 号

责任编辑： 柴文强
责任校对： 徐俊伟
责任印制：
出版发行： 清华大学出版社　　**地　址：** 北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编：** 100084
社 总 机： 010-62770175　　**邮　购：** 010-62786544
投稿与读者服务： 010-62795954，jsjjc@tup.tsinghua.edu.cn
质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　销： 全国新华书店
开　本： 185×230　**印　张：** 41.25　**防伪页：** 11　**字　数：** 909 千字
版　次： 2011 年 9 月第 1 版　**印　次：** 2011 年 9 月第 1 次印刷
印　数：
定　价： 元

产品编号：043949-01

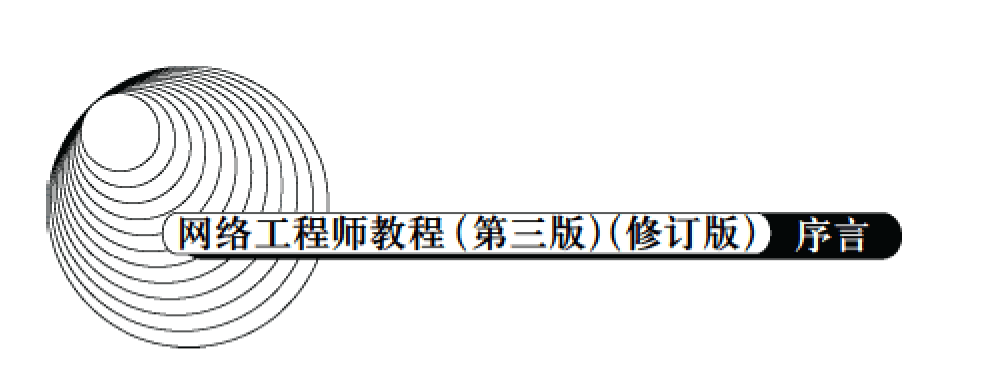
序　言

软件产业是信息产业的核心之一，是经济社会发展的基础性、先导性和战略性产业，在推进信息化与工业化融合、促进发展方式转变和产业结构升级、维护国家安全等方面有着重要作用。党中央、国务院高度重视软件产业发展，先后出台了 18 号文件、47 号文件等一系列政策措施，营造了良好的发展环境。近年来，我国软件产业进入快速发展期。2007 年销售收入达到 5834 亿元，出口 102.4 亿美元，软件从业人数达 148 万人。全国共认定软件企业超过 1.8 万家，登记备案软件产品超过 5 万个。软件技术创新取得突破，国产操作系统、数据库、中间件等基础软件相继推出并得到了较好的应用。软件与信息服务外包蓬勃发展，软件正版化工作顺利推进。

随着软件产业的快速发展，软件人才需求日益迫切。为适应产业发展需求、规范软件专业人员技术资格，20 余年前全国计算机软件考试创办，率先执行了以考代评政策。近年来，考试作了很多积极的探索，进行了一系列改革，考试名称、考试内容、专业类别、职业岗位也作了相应的变化。目前，考试名称已调整为计算机技术与软件专业技术资格（水平）考试，涉及 5 个专业类别、3 个级别层次共 27 个职业岗位，采取水平考试的形式，执行资格考试政策，并扩展到高级资格，取得了良好效果。20 余年来，累计报考人数近 200 万，影响力不断扩大。程序员、软件设计师、系统分析师、网络工程师、数据库系统工程师的考试标准已与日本相应考试级别实现互认，程序员和软件设计师的考试标准与韩国实现互认。通过考试，一大批软件人才脱颖而出，为加快培育软件人才队伍、推动软件产业健康发展起到了重要作用。

最近，工业和信息化部电子教育与考试中心组织了一批具有较高理论水平和丰富实践经验的专家编写了这套全国计算机技术与软件专业技术资格（水平）考试教材和辅导用书。按照考试大纲的要求，教材和辅导用书全面介绍相关知识与技术，帮助考生学习备考，将为软件考试的规范和完善起到积极作用。

我相信，通过社会各界共同努力，全国计算机技术与软件专业技术资格（水平）考试将



更加规范、科学，培养出更多专业技术人才，为加快发展信息产业、推动信息化与工业化融合做出积极贡献。

工业和信息化部副部长 奚国华

前　言

根据新的网络工程师考试大纲，这次修订版时对本书内容进行了比较大的调整。对基础知识部分进行了简化，对应用技术部分进行了改写，突出了网络服务器的配置、路由器和交换机的配置、以及网络安全和网络管理等实用技术。在适当调整后，全书缩减为 10 章，由雷震甲、吴晓葵、严体华编写，主要内容介绍如下。

第 1 章介绍计算机网络的基本概念，主要内容是计算机网络的体系结构——ISO 开放系统互连参考模型及基本概念，例如协议实体、协议数据单元，服务数据单元、面向连接的服务和无连接的服务、服务原语、服务访问点、相邻层之间的多路复用，以及各个协议层的功能特性等，都是进行网络分析的理论基础，是网络工程技术人员应该掌握的基础知识。

第 2 章讲述数据通信的基础知识，主要为物理层的内容。网络工程师除了熟悉网络协议的工作原理、能够操作网络互连设备之外，也应该掌握数据通信方面的基础知识，这样，在进行网络故障分析和故障排除时才能做到有的放矢，事半功倍地解决问题。

第 3 章介绍电话网、数据通信网、帧中继网和综合业务数字网等广域通信网方面的基础知识，这些网络都是进行网络互连时必须要用到的基础设施，这方面的基础知识可以帮助网络工程师根据已有的条件选择网络互连设备。

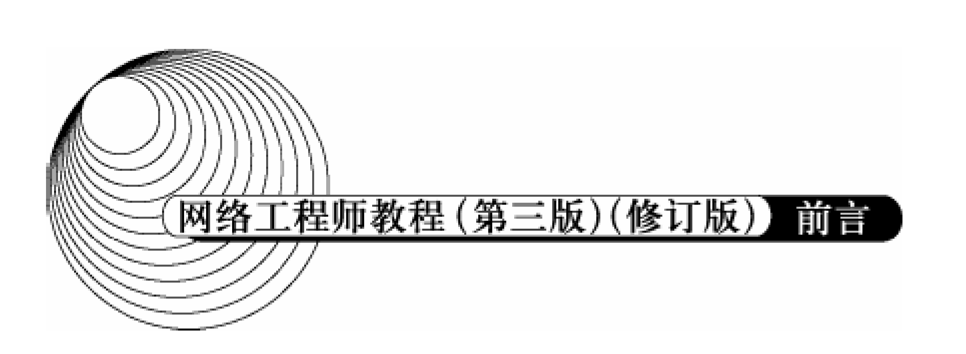
第 4 章详细介绍局域网和城域网方面的主要技术。这次修改时突出了快速以太网技术，删去了较少使用的令牌环网等，丰富了无线局域网和城域网方面的内容。这一章是网络工程师应该掌握的最重要的基础知识。

第 5 章讨论了网络互连的基本原理，深入讲解了 Internet 协议及其提供的网络服务。这一章也是网络工程师应该掌握的重要的基础知识。

第 6 章包含了网络安全方面的基础知识和应用技术。包括诸如数据加密、报文认证、数字签名等基本理论、网络安全协议的工作原理以及针对具体的网络系统设计和实现简单的安全解决方案。

第 7 章介绍了 Windows 和 Linux 操作系统的基础知识，并详细讲述了常用的各种服务器的配置方法。这一章的内容主要是在具体操作方面，网络工程师要能够熟练地配置各种网络服务器，排除网络服务器中出现的故障。

第 8 章是有关网络互连设备操作方面的基础知识和实用技术，这一章也是要求能够熟练地操作，重点是 VLAN 和动态路由配置。要求网络工程师能够熟悉网络互连设备的工作原理，掌



握路由器和交换机的配置命令，能够排除网络互连设备的故障。

第 9 章是网络管理，读者除了要熟悉 SNMP 协议的体系结构和操作原理之外，还要能实际操作网络管理系统，熟练地使用常见的网络管理命令，针对具体的网络给出实用的网络管理解决方案。

第 10 章讲述网络规划与设计。网络工程师应该能够根据网络的设计目标，按照系统工程的方法给出解决方案，写出规范的设计和实施文档。另外，这一章还给出了网络规划和设计的案例，作为学习时的参考。

新大纲增加了 IPv6、802.11x、MPLS、光纤主干网等新技术，希望读者给予注意。

编者　2011 年 7 月

目　录

第 1 章　计算机网络概论

计算机网络是计算机技术与通信技术相结合的产物。计算机网络是信息收集、分发、存储、处理和消费的重要载体。计算机网络作为一种生产和生活工具被人们广泛接纳和使用之后，对人类社会的经济、政治和文化生活产生了重大影响。本章讲述计算机网络的基本概念和发展简史，以及国际标准化组织定义的开放系统互连参考模型，后者是分析和认识计算机网络的理论基础。

1.1　计算机网络的形成和发展

1．早期的计算机网络

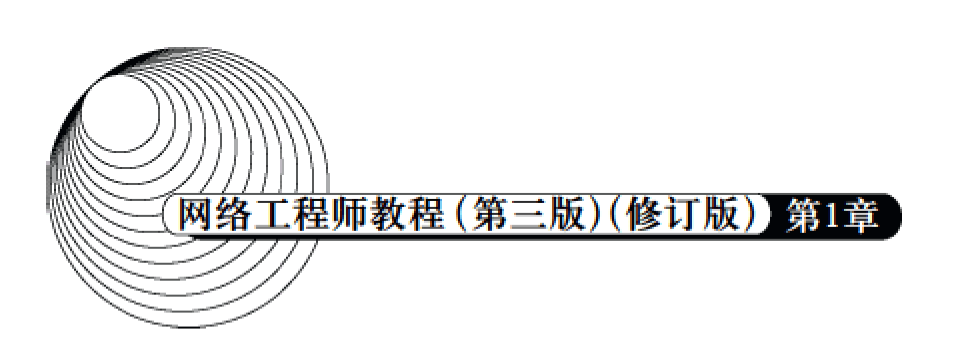
自从有了计算机，就有了计算机技术与通信技术的结合。早在 1951 年，美国麻省理工学院林肯实验室就开始为美国空军设计称为 SAGE 的半自动化地面防空系统，该系统最终于 1963 年建成，被认为是计算机和通信技术结合的先驱。

计算机通信技术应用于民用系统方面，最早的当数美国航空公司与 IBM 公司在 20 世纪 50 年代初开始联合研究、60 年代初投入使用的飞机订票系统 SABRE-I。美国通用电气公司的信息服务系统则是世界上最大的商用数据处理网络，其地理范围从美国本土延伸到欧洲、澳洲和亚洲的日本。该系统于 1968 年投入运行，具有交互式处理和批处理能力，由于地理范围大，可以利用时差达到资源的充分利用。

在这一类早期的计算机通信网络中，为了提高通信线路的利用率并减轻主机的负担，已经使用了多点通信线路、终端集中器以及前端处理机等现代通信技术。这些技术对以后计算机网络的发展有着深刻的影响。以多点线路连接的终端和主机间的通信建立过程，可以用主机对各终端轮询或是由各终端连接成雏菊链的形式实现。考虑到远程通信的特殊情况，对传输的信息还要按照一定的通信规程进行特别的处理。

2．现代计算机网络的发展

20 世纪 60 年代中期出现了大型主机，同时也出现了对大型主机资源远程共享的要求。以程控交换为特征的电信技术的发展则为这种远程通信需求提供了实现的手段。现代意义上的计算机网络是从 1969 年美国国防部高级研究计划局（DARPA）建成的 ARPAnet 实验网开始的。



该网络当时只有 4 个节点，以电话线路作为主干通信网络，两年后，建成 15 个节点，进入工作阶段。此后，ARPAnet 的规模不断扩大。到了 20 世纪 70 年代后期，网络节点超过 60 个，主机 100 多台，地理范围跨越了美洲大陆，连通了美国东部和西部的许多大学和研究机构，而且通过通信卫星与夏威夷和欧洲地区的计算机网络相互连通。

ARPAnet 的主要特点是：

（1）资源共享；

（2）分散控制；

（3）分组交换；

（4）采用专门的通信控制处理机；

（5）分层的网络协议。

这些特点被认为是现代计算机网络的一般特征。

20 世纪 70 年代中后期是广域通信网大发展的时期。各发达国家的政府部门、研究机构和电报电话公司都在发展分组交换网络。例如，英国邮政局的 EPSS 公用分组交换网络（1973）、法国信息与自动化研究所（IRIA）的 CYCLADES 分布式数据处理网络（1975）、加拿大的 DATAPAC 公用分组交换网（1976）以及日本电报电话公司的 DDX-3 公用数据网（1979）等。这些网络都以实现计算机之间的远程数据传输和信息共享为主要目的，通信线路大多采用租用电话线路，少数铺设专用线路，数据传输速率在 50Kb/s 左右。这一时期的网络被称为第二代网络，以远程大规模互连为其主要特点。

3．计算机网络标准化阶段

经过 20 世纪六七十年代前期的发展，人们对组网的技术、方法和理论的研究日趋成熟。为了促进网络产品的开发，各大计算机公司纷纷制定自己的网络技术标准。IBM 首先于 1974 年推出了该公司的系统网络体系结构（System Network Architecture，SNA），为用户提供能够互连互通的成套通信产品；1975 年，DEC 公司宣布了数字网络体系结构（Digital Network Architecture，DNA）；1976 年，UNIVAC 发布了分布式通信体系结构（Distributed Communication Architecture）。这些网络技术标准只是在一个公司范围内有效，遵从某种标准的、能够互连的网络通信产品只是同一公司生产的同构型设备。网络通信市场这种各自为政的状况使得用户在投资方向上无所适从，也不利于多厂商之间的公平竞争。1977 年，国际标准化组织（ISO）的 TC97 信息处理系统技术委员会 SC16 分技术委员会开始着手制定开放系统互连参考模型 OSI/RM。作为国际标准，OSI 规定了可以互连的计算机系统之间的通信协议，遵从 OSI 协议的网络通信产品都是所谓的“开放系统”。今天，几乎所有的网络产品厂商都声称自己的产品是开放系统，不遵从国际标准的产品逐渐失去了市场。这种统一的、标准化产品互相竞争的市场进一步促进了网络技术的发展。

4．微型机局域网的发展时期

20 世纪 80 年代初期出现了微型计算机，这种适合办公室环境和家庭使用的新机种对社会生活的各个方面都产生了深刻的影响。1972 年，Xerox 公司发明了以太网，以太网与微型机的结合使得微型机局域网得到了快速的发展。在一个单位内部的微型计算机和智能设备互相连接起来，提供了办公自动化的环境和信息共享的平台。1980 年 2 月，IEEE 组织了一个 802 委员会，开始制定局域网标准。局域网的发展道路不同于广域网，局域网厂商从一开始就按照标准化、互相兼容的方式展开竞争。用户在建设自己的局域网时选择面更宽，设备更新更快。

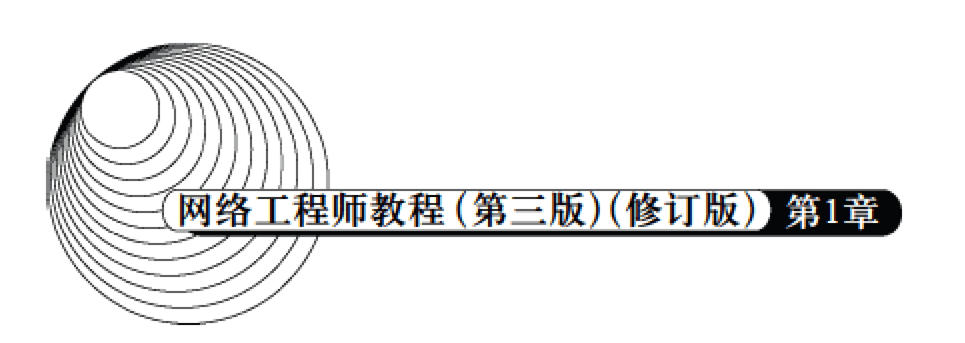
5．国际因特网的发展时期

1985 年，美国国家科学基金会（National Science Foundation，NSF）利用 ARPAnet 协议建立了用于科学研究和教育的骨干网络 NSFnet。1990 年，NSFnet 代替 ARPAnet 成为美国国家骨干网，并且走出了大学和研究机构进入社会。从此，网上的电子邮件、文件下载和消息传输受到越来越多人的欢迎并被广泛使用。1992 年，Internet 学会成立，该学会把 Internet 定义为“组织松散的、独立的国际合作互联网络”，“通过自主遵守计算协议和过程支持主机对主机的通信”。1993 年，美国伊利诺斯大学国家超级计算中心开发成功了网上浏览工具 Mosaic（后来发展成 Netscape），使得各种信息都可以方便地在网上交流。浏览工具的实现引发了 Internet 发展和普及高潮。上网不再是网络操作人员和科学研究人员的专利，而成为一般人进行远程通信和交流的工具。在这种形势下，美国总统克林顿于 1993 年宣布正式实施国家信息基础设施（National Information Infrastructure，NII）计划，从此在世界范围内展开了争夺信息化社会领导权和制高点的竞争。与此同时，NSF 不再向 Internet 注入资金，使其完全进入商业化运作。20 世纪 90 年代后期，Internet 以惊人的高速度发展，网上的主机数量、上网人数、网络的信息流量每年都在成倍地增长。

1.2 计算机网络的分类和应用

1.2.1 计算机网络的分类

“计算机网络”这一术语是指由通信线路互相连接的许多自主工作的计算机构成的集合体。这里强调构成网络的计算机是自主工作的，这是为了和多终端分时系统相区别。在后一种系统中，终端无论是本地的还是远程的，只是主机和用户之间的接口，它本身并不拥有计算资源，全部资源集中在主机中。主机以自己拥有的资源分时地为各终端用户服务。计算机网络中的各个计算机（工作站）本身拥有计算资源，能独立工作，能完成一定的计算任务。同时，用户还



可以共享网络中其他计算机的资源（CPU、大容量外存或信息等）。

比计算机网络更高级的系统是分布式系统。分布式系统在计算机网络基础上为用户提供了透明的集成应用环境。用户可以用名字或命令调用网络中的任何资源或进行远程的数据处理，而不必考虑这些资源或数据的地理位置。

与计算机网络类似的另一种系统是多机系统。多机系统专指同一机房中的许多大型主机互连组成的、能高速并行处理的计算机系统。对这种系统互连的要求是高带宽和连通的多样性。计算机网络中的信息传输开销很大，实际的有效数据速率比通信线路能够提供的带宽要小得多。同时由于距离的原因，计算机网络终端系统是通过交换设备互连的，这种有限互连的方式不能适应高速并行计算的要求。

计算机网络的组成元素可以分为两大类，即网络节点和通信链路。网络节点又分为端节点和转发节点。端节点指信源和信宿节点，例如用户主机和用户终端；转发节点指网络通信过程中控制和转发信息的节点，例如交换机、集线器、接口信息处理机等。通信链路是指传输信息的信道，可以是电话线、同轴电缆、无线信道、卫星线路、微波中继线路和光纤缆线等。网络节点通过通信链路连接成的计算机网络如图 1-1 所示。

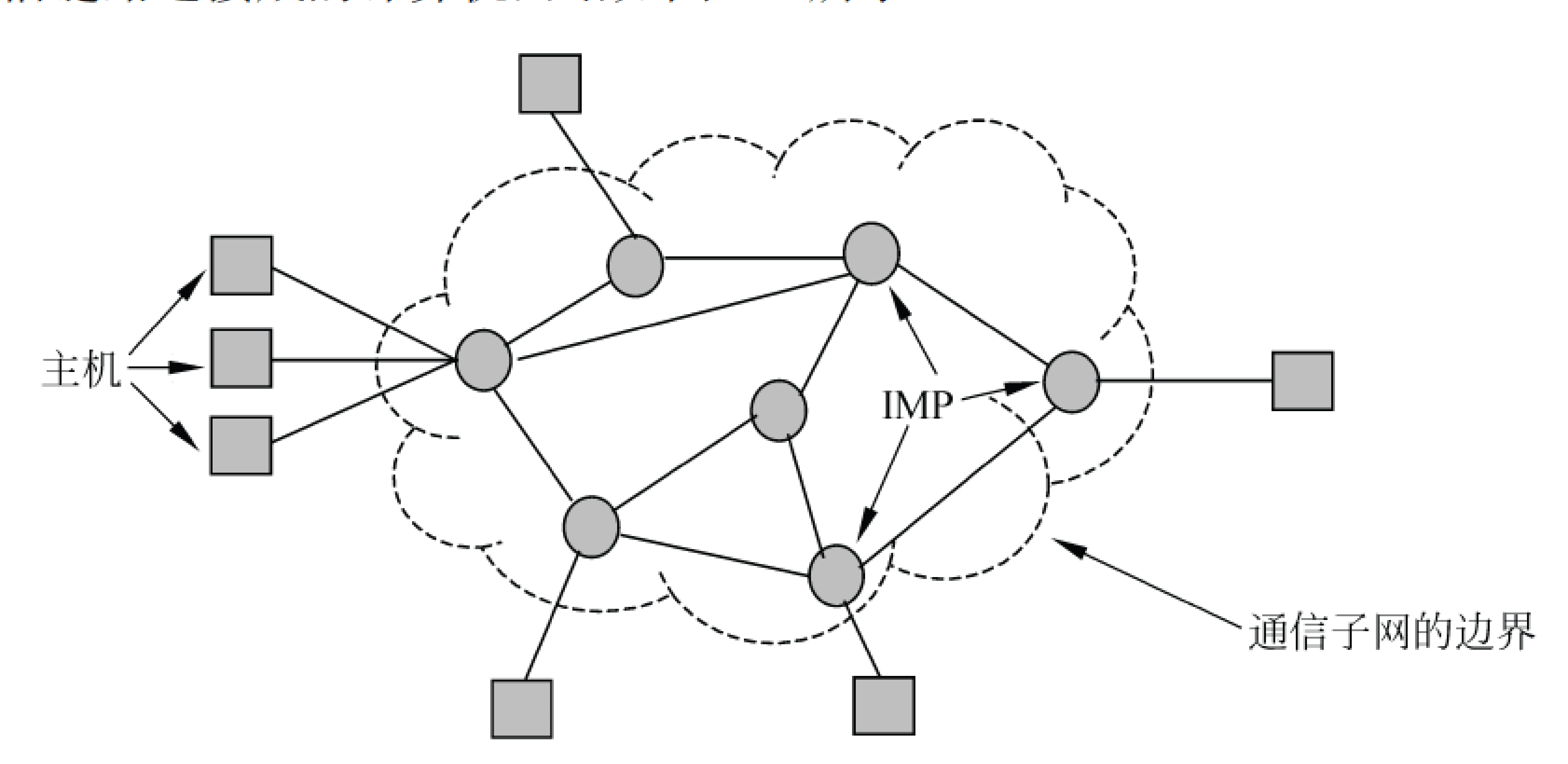


图 1-1　通信子网与资源子网

在图 1-1 中，虚线框外的部分称为资源子网。资源子网中包括拥有资源的用户主机和请求资源的用户终端，它们都是端节点。虚线框内的部分叫做通信子网，其任务是在端节点之间传送由信息组成的报文，主要由转发节点和通信链路组成。在图 1-1 中，按照 ARPA 网络的术语把转发节点通称为接口信息处理机（Interface Message Processor，IMP）。IMP 是一种专用于通信的计算机，有些 IMP 之间直接相连，有些 IMP 之间必须经过其他 IMP 间接相连。当 IMP 收到一个报文后要根据报文的目标地址决定把该报文提交给与它相连的主机还是转发到下一个 IMP，这种通信方式叫做存储-转发通信。在广域网中的通信一般都采用这种方式。另外一种通信方式是广播通信方式，主要用于局域网中。局域网中的 IMP 简化为一个微处理器芯片，每台

- 模块化交换机。这种交换机的机箱中预留了一定数量的插槽，用户可以根据网络扩充的需求选择不同类型的端口模块。这种交换机具有更大的可扩充性。

（4）根据配置方式划分。

- 堆叠型交换机。这种交换机具有专门的堆叠端口，用堆叠电缆把一台交换机的 UP 口连接到另一台交换机的 DOWN 口，以实现端口数量的扩充（如图 8-1 所示）。一般交换机能够堆叠 4～9 层，所有交换机可以当作一台交换机来统一管理。
- 非堆叠型交换机。这种交换机没有堆叠端口，但可以通过级连方式进行扩充。级连模式使用以太网端口（100M FE 端口、GE 端口或 10GE 端口）进行层次间互联（如图 8-2 所示），可以通过统一的网管平台实现对全网设备的管理。为了保证网络运行的效率，级连层数一般不要超过 4 层。

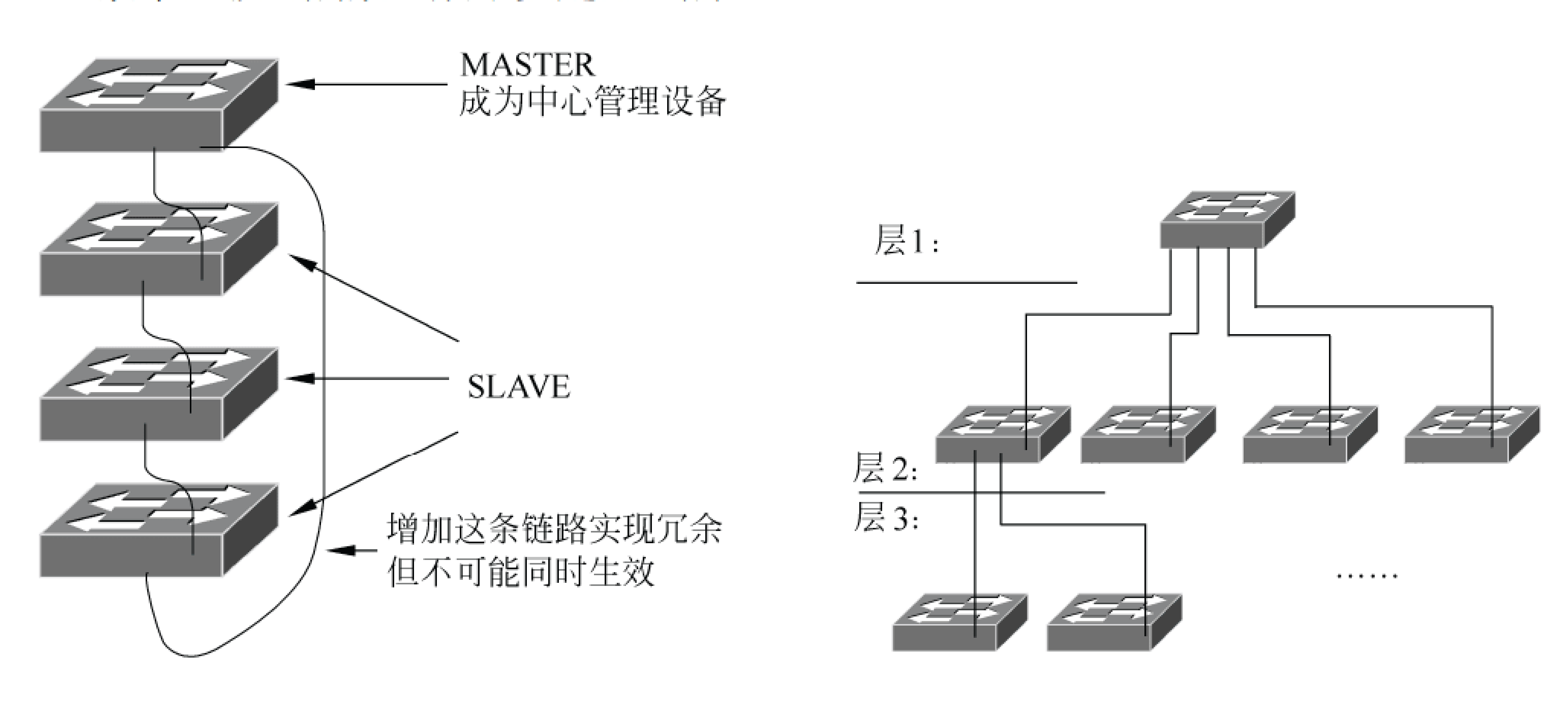


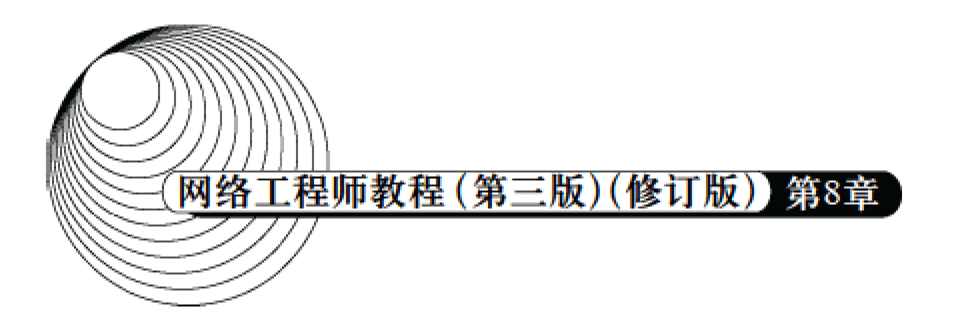
图 8-1　交换机的堆叠　　　　图 8-2　交换机的级连

（5）根据管理类型划分。

- 网管型交换机。这种交换机支持简单网络管理协议（SNMP）和管理信息库（MIB），可以指定 IP 地址，实现远程配置、监视和管理。
- 非网管型交换机。这种交换机不支持 SNMP 和 MIB，只能根据 MAC 地址进行交换，无法进行功能配置和管理。
- 智能型交换机。这种交换机支持基于 Web 的图形化管理和 MIB-II，无须使用复杂的命令行管理方式，配置和维护比较容易。更重要的是，智能型交换机提供 QoS 管理、VPN、用户认证以及多媒体传输等复杂的应用功能，而不仅是转发数据分组。

（6）根据适用范围划分。

网络的分层结构把复杂的大型网络分解为多个容易管理的小型网络，每一层交换设备分别



（2）传输模式。

- 半双工（half-duplex）。半双工交换机在一个时间段内只能有一个动作发生，发送和接收不能同时进行。早期的集线器是半双工产品，随着技术进步，半双工方式逐渐被淘汰。
- 全双工（full-duplex）。全双工交换机在发送数据的同时也能接收数据，两者同步进行。全双工传输需要使用两对双绞线或两根光纤，一般双绞线端口和光纤端口都支持全双工传输模式。这种传输模式在一对主机之间建立了一条虚拟的专用连接，使得数据速率成倍提高。
- 全双工/半双工自适应。在以上两种方式之间可以自动切换。1000Base-TX 支持自适应，而 1000Base-SX、1000Base-LX、1000Base-LH 和 1000Base-ZX 均不支持自适应，不同速率和传输模式的光纤端口间无法进行通信，因而要求相互连接的光纤端口必须具有完全相同的传输速率和传输模式，否则将导致连通故障。千兆光纤端口标准如表 8-1 所示。

表 8-1 千兆光纤端口标准

标 准	波长（nm）	光 纤 类 型	最大传输距离
1000Base-SX (Short-wave)	850	62.5/125um 多模光纤 50/125um 多模光纤	220m 500m
1000Base-LX (long-wave)	1310	62.5/125um 多模光纤 50/125um 多模光纤 9/125um 单模光纤	550m 550m 10km
1000Base-LH (long-haul)	1310	9/125um 单模光纤	40km
1000Base-ZX (extended range)	1550	9/125um 单模光纤	50km 或 80km

（3）包转发率。包转发率也称端口吞吐率，指交换机进行数据包转发的能力，单位为 pps（package per second）。包转发速率是以单位时间内发送 64 字节数据包的个数作为计算基准的。对于千兆以太网来说，计算方法如下：

1000Mbps÷8b÷（64＋8＋12）byte=1 488 095pps

当以太网帧为 64 字节时，需考虑 8 字节的帧头和 12 字节的帧间隙开销。据此，包转发速率的计算方法如下：

包转发率=千兆端口数×1.488Mpps＋百兆端口数×0.1488Mpps＋其余端口×相应计算方法

（4）背板带宽。交换机的背板带宽是指交换机端口处理器和数据总线之间单位时间内所能传输的最大数据量。背板带宽标志了交换机总的交换能力，单位为 Gbps。一般交换机的背板带宽从几个 Gbps 到上千个 Gbps。交换机所有端口能提供的总带宽的计算公式为：

总带宽＝端口数×端口速率×2（全双工模式）

如果总带宽小于标称背板带宽，那么可以认为背板带宽是线速的。例如，Catalyst 6500 系列交换机的背板带宽可扩展到 256Gbps，包转发速率可扩展到 150Mpps。

（5）MAC 地址数。交换机可以识别网络节点的 MAC 地址，并把它放到 MAC 地址表中。MAC 地址表存放在交换机的缓存中，当需要向目标地址发送数据时，交换机就在 MAC 地址表中查找相应 MAC 地址的节点位置，然后直接向这个位置的节点转发。MAC 地址数是指交换机的 MAC 地址表中可以存储的 MAC 地址数量。

不同档次的交换机端口所能够支持的 MAC 地址数量不同。在交换机的每个端口，都需要足够的缓存来记忆这些 MAC 地址，所以缓存容量的大小决定了交换机所能记忆的 MAC 地址数。

（6）VLAN 表项。VLAN 是一个独立的广播域，可有效地防止广播风暴。由于 VLAN 基于逻辑连接而不是物理连接，因此配置十分灵活。在有第三层交换功能的基础上，VLAN 之间也可以通信。最大 VLAN 数量反映了一台交换机所能支持的最大 VLAN 数目。目前交换机 VLAN 表项数目在 1024 以上，可以满足一般企业的需要。

（7）机架插槽数。固定配置不带扩展槽的交换机仅支持一种类型的网络，固定配置带扩展槽的交换机和机架式交换机可支持一种以上类型的网络，例如以太网、快速以太网、千兆以太网、ATM 网、令牌环网及 FDDI 等。一台交换机所支持的网络类型越多，可扩展性就越强。机架插槽数是指机架式交换机所能安插的最大模块数，扩展槽数是指固定配置带扩展槽的交换机所能安插的最大模块数。

3. 交换机支持的以太网协议

有关交换机的以太网协议如表 8-2 所示。

表 8-2 交换机支持的以太网协议

标　准	说　明	规　范
IEEE 802.3i	以太网 10Base-T 规范	两对 UTP，RJ-45 连接器，传输距离 100m
IEEE 802.3u	快速以太网物理层规范	100Base-TX：两对 5 类 UTP，支持 10Mbps、100Mbps 自动协商。 100Base-T4：四对 3 类 UTP。 100Base-FX：光纤
IEEE 802.3z	千兆以太网物理层规范	1000Base-SX：短波 SMF。 1000Base-LX：长波 SMF 或 MMF
IEEE 802.3ab	双绞线千兆以太网物理层规范	1000Base-TX
IEEE 802.3ad	Link Aggregation Control Protocol（LACP）	链路汇聚技术可以将多个链路绑定在一起，形成一条高速链路，以达到更高的带宽，并实现链路备份和负载均衡

（2）AUI 端口。AUI 端口是一种 D 型 15 针连接器，用在令牌环网或总线型以太网中。路由器经 AUI 端口通过粗同轴电缆收发器连接 10Base-5 网络，也可以通过外接的 AUI-to-RJ-45 适配器连接 10Base-T 以太网，还可以借助其他类型的适配器实现与 10Base-2 细同轴电缆或 10Base-F 光缆的连接。AUI 端口如图 8-9 所示。

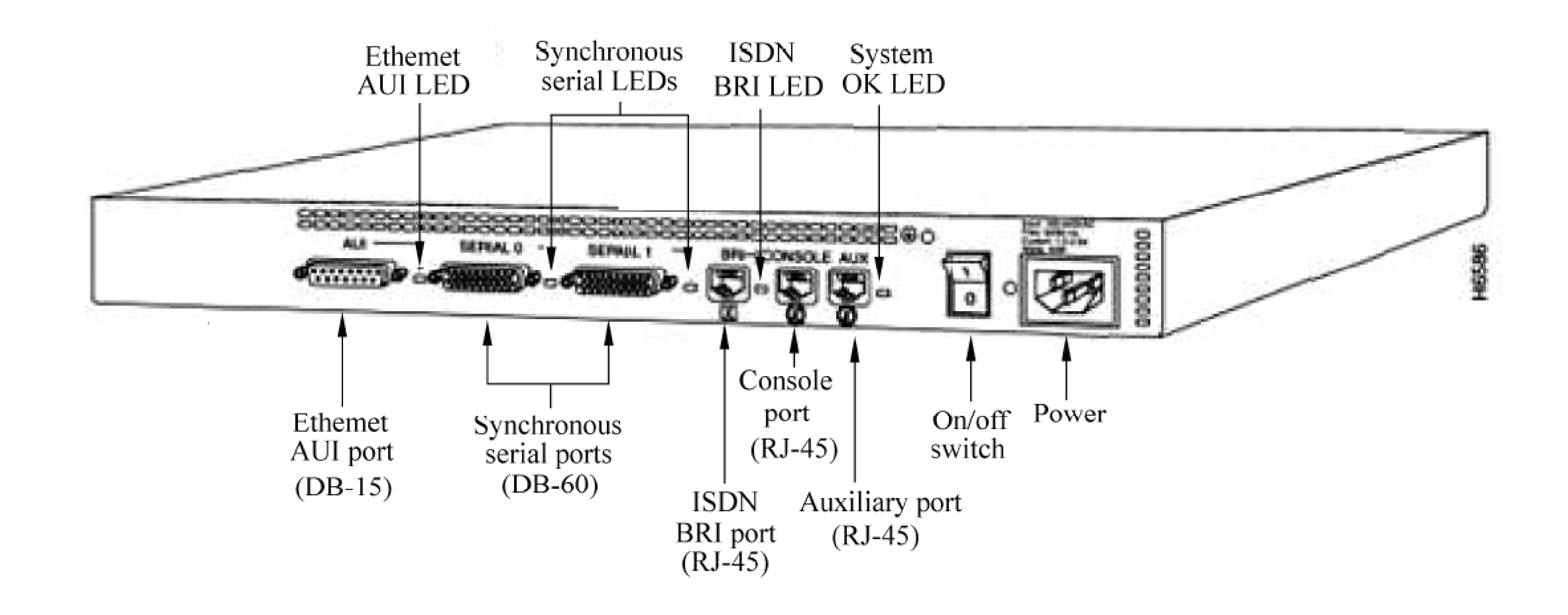


图 8-9 路由器背板示意图

（3）高速同步串口。在路由器与广域网的连接中，应用最多的是高速同步串行口（Synchronous Serial Port），这种端口用于连接 DDN、帧中继、X.25 和 PSTN 等网络。通过这种端口所连接的网络两端要求同步通信，以很高的速率进行数据传输。高速同步串行口如图 8-9 所示。

（4）ISDN BRI 端口。ISDN BRI 端口（如图 8-9 所示）通过 ISDN 线路实现路由器与 Internet 或其他网络的远程连接。ISDN BRI 三个通道（2B+D）的总带宽为 144 Kbps，端口采用 RJ-45 标准，与 ISDN NT1 的连接使用 RJ-45-to-RJ-45 直通线。

（5）异步串口。异步串口（ASYNC）主要应用于与 Modem 或 Modem 池的连接，以实现远程计算机通过 PSTN 拨号接入。异步端口的速率不是很高，也不要求同步传输，只要求能连续通信就可以了。图 8-10 所示为异步串口。

（6）Console 端口。Console 端口通过配置专用电缆连接至计算机串行口，利用终端仿真程序（如 Windows 中的超级终端）对路由器进行本地配置。路由器的 Console 端口为 RJ-45 口（如图 8-9 所示）。Console 端口不支持硬件流控。

（7）AUX 端口。对路由器进行远程配置时要使用 AUX 端口（Auxiliary Prot），如图 8-9 所

示。AUX 端口在外观上与 RJ-45 端口一样，只是内部电路不同，实现的功能也不一样。通过 AUX 端口与 Modem 进行连接必须借助 RJ-45 to DB9 或 RJ-45 to DB25 适配器进行电路转换。AUX 端口支持硬件流控。

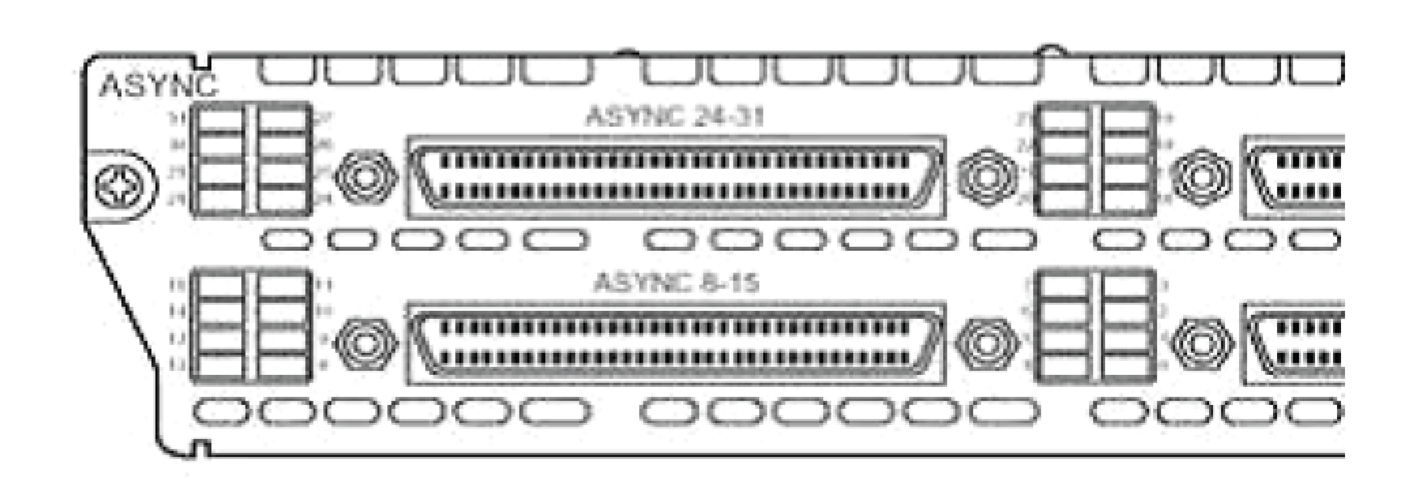


图 8-10　异步串口

3．路由器的操作系统

一般的路由器都有一个操作系统，各个厂家的路由器操作系统不尽相同，但都以 Cisco 的因特网操作系统（Internetwork Operating System，IOS）作为工作标准。熟悉了 Cisco IOS 的操作，对其他路由器操作系统也不难掌握。

每种路由器平台的 IOS 版本都不同，事实上有几百个不同的 IOS 版本，甚至会有一些特定版本的 IOS 提供一些特殊的功能和解决方案，例如适合服务提供商的 IOS、适合企业的 IOS 或者适合 SNA 集成或支持 IPX 的 IOS 等。Cisco IOS 一般有几兆字节大小，运行在路由器或交换机上，为这些交换设备提供一个管理平台，确保网络的连通性、可靠性、安全性、服务质量和可伸缩性等性能指标。

路由器或交换机的操作是由配置文件（configuration file 或 config）控制的。配置文件包含有关设备如何操作的指令，是由网络管理员创建的，一般有几百到几千个字节大小。

IOS 的每一个组件都是作为独立的文件存放在存储器中，不同类型的存储器见下面的介绍。

IOS 命令在所有路由器产品中都是通用的。这意味着只要掌握一个操作界面就可以了，即命令行界面（Command Line Interface，CLI）。所以无论是通过控制台端口，或通过一部 Modem，还是通过 Telnet 连接来配置路由器，看到的命令行界面都是相同的。

IOS 有三种命令模式，即用户模式（User mode）、特权模式（Privileged mode）和配置模式（Configuration mode）。在不同的命令模式中可执行的命令集不同，可实现的管理功能也不同，详见下面的解释。

8.1.3　访问路由器和交换机

要对网络互连设备进行具体的配置首先就要有效地访问它们，一般来说可以用以下几种方

法访问路由器或交换机。

（1）通过设备的 Console（控制台）端口接终端或运行终端仿真软件的计算机。

（2）通过设备的 AUX 端口接 Modem，通过电话线与远方的终端或运行终端仿真软件的计算机相连。

（3）通过 Telnet 程序。

（4）通过浏览器访问。

（5）通过网管软件。

下面以路由器为例给出几种访问网络互连设备方法的连接图（如图 8-11 所示）。

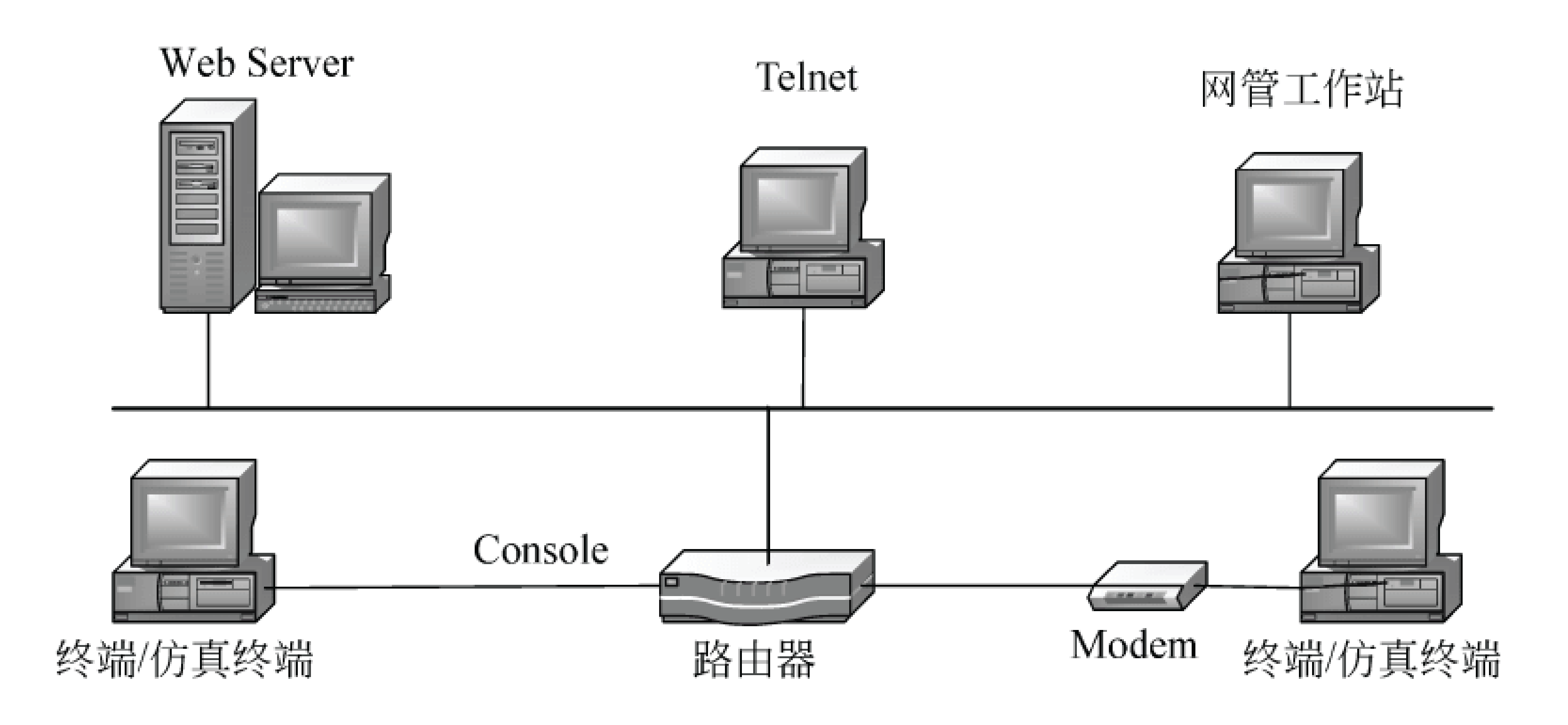


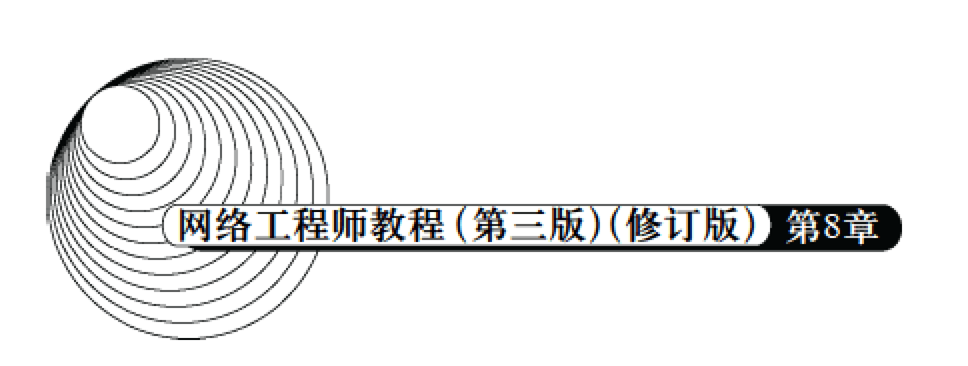
图 8-11　访问路由器的几种方法

但是，路由器的第一次设置必须通过第一种方法来实现，同时第一种方法也是最常用、最直接有效的一种配置方法。因此，本书中对路由器和交换机的配置都是通过 Console 端口连接运行超级终端仿真软件的 PC 来实现。

Console 端口是路由器和交换机设备的基本端口，它是对一台新的路由器和交换机进行配置时必须使用的接口。连接 Console 端口的线称为控制台电缆（Console Cable）。在具体的连接上，Console 电缆一端插入网络设备的 Console 端口，另一端接入终端或 PC 的串行接口，从而实现对设备的访问和控制。

8.2　交换机的配置

不同厂家生产的不同型号的交换机，其具体的配置命令和方法是有差别的。不过配置的原理基本都是相同的，本节中主要以 Cisco Crystal 2950 系列交换机为例讲解交换机配置的基本技术和技能。




```
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(13)EA1, RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Tue 04-Mar-03 02:14 by yenanh
Image text-Base: 0x80010000, data-Base: 0x805A8000

Initializing flashfs...                         （初始化 Flash 文件系统）
flashfs[1]: 18 files, 2 directories
flashfs[1]: 0 orphaned files, 0 orphaned directories
flashfs[1]: Total bytes: 7741440
flashfs[1]: Bytes used: 4871168
flashfs[1]: Bytes available: 2870272
flashfs[1]: flashfs fsck took 7 seconds.
flashfs[1]: Initialization complete.
Done initializing flashfs.
POST: System Board Test : Passed                （系统板自检）
POST: Ethernet Controller Test : Passed         （以太网控制器自检）
ASIC Initialization Passed                      （专用芯片自检）

POST: FRONT-END LOOPBACK TEST : Passed          （环路自检）
cisco WS-C2950-24 (RC32300) processor (revision J0) with 20839K bytes of memory.

Processor board ID FOC0718Y0EA
Last reset from system-reset
Running Standard Image                          （软件版本为标准版）
24 FastEthernet/IEEE 802.3 interface(s)

32K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 00:0D:28:C0:12:40
（以下为各部件号、序列号及版本号）
Motherboard assembly number: 73-5781-11
Power supply part number: 34-0965-01B0
Motherboard serial number: FOC061903RT
Power supply serial number: PHI0714070Y
Model revision number: J0
Motherboard revision number: A0
```

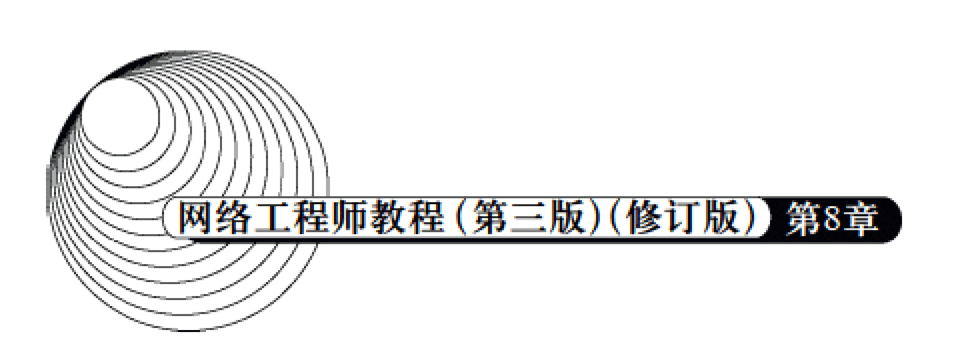
```
C2950(config)#ip domain-name cisco.com          （设置域名）
C2950(config)#ip name-server 200.0.0.1          （设置域名服务器）
C2950(config)#end
```

（3）配置交换机的端口属性。交换机的端口属性默认地支持一般网络环境下的正常工作，一般情况下是不需要对其端口进行设置的。在某些情况下需要对其端口属性进行配置时，配置的对象主要有速率、双工和端口描述等信息。

```
C2950(config)#interface fastethernet0/1        （进入接口 0/1 的配置模式）
C2950(config-if)#speed ?                       （查看 speed 命令的子命令）
    10     Force 10 Mbps operation             （显示结果）
    100    Force 100 Mbps operation
    auto Enable AUTO speed configuration
C2950(config-if)#speed 100                     （设置该端口速率为 100Mbps）
C2950(config-if)#duplex ?                      （查看 duplex 命令的子命令）
    auto Enable AUTO duplex configuration
    full   Force full duplex operation
    half Force half-duplex operation
C2950(config-if)#duplex full                   （设置该端口为全双工）
C2950(config-if)#description TO_PC1            （设置该端口描述为 TO_PC1）
C2950(config-if)#^Z                            （返回到特权模式，同 end）
C2950#show interface fastethernet0/1           （查看端口 0/1 的配置结果，结果略）
C2950#show interface fastethernet0/1 status    （查看端口 0/1 的状态，结果略）
```

（4）配置和查看 MAC 地址表。有关 MAC 地址表的配置有三个方面，即超时时间、永久地址和限制性地址。交换机学习到的动态 MAC 地址的超时时间默认为 300s，可以通过命令来修改这个值。设置了静态 MAC 地址，这个地址永久存在于 MAC 地址表中，不会超时，所有端口均可以转发以太网帧给该端口。限制性静态（restricted static）地址是在永久地址的基础上，同时限制了源端口，其安全性更高。

```
C2950(config)#mac-address-table ?                     （查看 mac-address-table 的子命令）
    aging-time       Aging time of dynamic addresses
    permanent  Configure a permanent address
    restricted    Configure a restricted static address
C2950(config)#mac-address-table aging-time 100        （设置超时时间为 100s）
C2950(config)#mac-address-table permanent 0000.0c01.bbcc f0/3        （加入永久地址）
C2950(config)#mac-address-table restricted static 0000.0c02.bbcc f0/6 f0/7    （加入限制静态地址）
C2950(config)#end
C2950#show mac-address-table          （查看整个 MAC 地址表）
```



交换机的初始状态是工作在透明模式，有一个默认的 VLAN，所有的端口都属于这个 VLAN。

1．划分 VLAN 的方法

虚拟局域网是交换机的重要功能，通常虚拟局域网的实现形式有三种，即静态端口分配、动态虚拟网和多虚拟网端口配置。

静态虚拟网的划分通常是网管人员使用网管软件或直接设置交换机的端口，使其直接从属某个虚拟网。这些端口一直保持这些从属性，除非网管人员重新设置。这种方法虽然比较麻烦，但比较安全，容易配置和维护。

支持动态虚拟网的端口，可以借助智能管理软件自动确定它们的从属。端口是通过借助网络包的 MAC 地址、逻辑地址或协议类型来确定虚拟网的从属。当一个网络节点刚连接入网时，交换机端口还未分配，于是交换机通过读取网络节点的 MAC 地址动态地将该端口划入某个虚拟网。这样，一旦网管人员配置好后，用户的计算机可以灵活地改变交换机端口，而不会改变该用户的虚拟网从属性。

多虚拟网端口配置支持一个用户或一个端口可以同时访问多个虚拟网。这样可以将一台网络服务器配置成多个业务部门（每种业务设置成一个虚拟网）都可同时访问，也可以同时访问多个虚拟网的资源，还可让多个虚拟网间的连接只需一个路由端口即可完成。但这样会带来安全上的隐患。

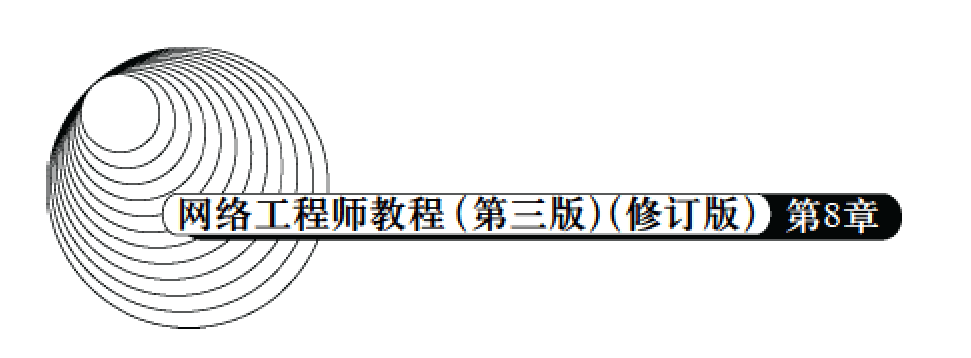
静态虚拟网是最普遍使用的一种划分 VLAN 的方法，下面就以该方法为例介绍 VLAN 配置的知识。

2．配置 VTP 协议

为了让读者清楚地了解 VTP（VLAN Trunking Protocol）协议的工作情况以及如何来配置 VTP 协议，结合一个综合实例，如图 8-15 所示的拓扑结构来配置 VTP 协议及跨交换机的 VLAN。用交叉双绞线把 2950A 交换机的 FastEthernet0/24 端口和 2950B 交换机的 FastEthernet0/24 端口连接起来，作为两交换机间的 Trunk 线路。

配置 2950A 交换机为服务器模式。

```
Switch>enable                        （进入特权模式）
Switch#config terminal               （进入配置子模式）
Switch(config)#hostname 2950A        （修改主机名为 2950A）
2950A(config)#end
2950A#
2950A #vlan dataBase                 （进入 VLAN 配置子模式）
```




```
0VTP V2 Mode                        : Disabled
VTP Traps Generation                : Disabled
MD5 digest                          : 0x82 0x6B 0xFB 0x94 0x41 0xEF 0x92 0x30
Configuration last modified by 0.0.0.0 at 3-1-93 00:02:51
2950A #
```

配置 2950B 交换机为客户端模式，则它会从服务器（2950A）那里学习到 VTP 的其他信息及 VLAN 信息。

```
Switch#config terminal                  （进入配置子模式）
Switch(config)#hostname 2950B           （修改主机名为 2950B）
2950B(config)#end
2950B#vlan dataBase
2950B(vlan)#vtp client
Setting device to VTP CLIENT mode.
2950B(vlan)#exit
```

3．配置 VLAN Trunk 端口

跨交换机的同一 VLAN 内的数据经过 Trunk 线路进行交换，默认情况下 trunk 允许所有的 VLAN 通过。可以使用 **switchport trunk allowed vlan remove** *vlan-list* 来去掉某一 VLAN。可以在交换机 2950A 和 2950B 上做如下相同的配置操作。

```
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.   End with CNTL/Z.
Switch(config)#interface f0/24                          （进入端口 24 配置模式）
Switch(config-if)#switchport mode trunk                 （设置当前端口为 Trunk 模式）
Switch(config-if)# switchport trunk allowed vlan all    （设置允许从该端口交换数据的 VLAN）
Switch(config-if)#exit
Switch(config)#exit
Switch#
```

4．创建 VLAN

VLAN 信息可以在服务器模式或透明模式交换机上创建。这里在 2950A 交换机上创建两个 VLAN。

```
2950A#vlan dataBase
2950A (vlan)#vlan 2                     （创建一个 VLAN2）
```

```
VLAN 2 added:
    Name: VLAN0002              （系统自动命名）
2950A (vlan)#vlan 3 name vlan3  （创建一个 VLAN3，并命名为 vlan3）
VLAN 3 added:
    Name: vlan3
2950A (vlan)#exit
```

5．将端口加入到某个 VLAN 中

配置完 VTP 协议及 VLAN Trunk 端口后就可以设置将端口归属于哪个 VLAN。在交换机 2950A 和 2950B 上做如下相同的配置操作，则 Vlan2 中包含两个交换机的 fa0/9 端口，Vlan3 中包含两个交换机的 f0/10 端口，其余端口可以做类似设置。除了加入 Vlan2 和 Vlan3 的端口外，其余各端口均属于 Vlan1（交换机默认的 VLAN）。

```
Switch#config termilal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface f0/9                    （进入端口 9 的配置模式）
Switch(config-if)#switchport mode access         （设置端口为静态 VLAN 访问模式）
Switch(config-if)#switchport access vlan 2       （把端口 9 分配给相信的 VLAN2）
Switch(config-if)#exit
Switch(config)#interface f0/10
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exit
Switch(config)#exit
Switch#show vlan                                 （查看 VLAN 配置信息）
（结果省略）
Switch#
```

8.2.4 生成树协议配置

生成树协议是交换式以太网中的重要概念和技术，该协议的目的是在实现交换机之间冗余连接的同时，避免网络环路的出现，实现网络的高可靠性。它通过在交换机之间传递 BPDU（Bridge Protocol Data Unit，桥接协议数据单元）来互相告知诸如交换机的桥 ID、链路性质和根桥 ID 等信息，以确定根桥，决定哪些端口处于转发状态，哪些端口处于阻断状态，以免引起网络环路。

当交换机之间有多个 VLAN 时 Trunk 线路负载会过重，这时需要设置多个 Trunk 端口，但这样会形成网络环路。而 STP 协议便可以解决这一问题。

交换机就可以学习到 Switch1 交换机上的 VLAN 信息，可以用 show vlan 命令来验证 Switch2 交换机是否学习到了 VLAN 信息。配置完 Switch2 交换机的 VTP 和 Trunk 以后，又回到 Switch1 交换机上来配置 STP。

```
（配置 STP 权值）
Switch1#config Terminal
Switch1(config)#interface f0/23                                （进入端口 23 配置模式，Trank1）
Switch1(config-if)# spanning-tree vlan 1 port-priority 10      （将 VLAN 1 的端口权值设为 10）
Switch1(config-if)# spanning-tree vlan 2 port-priority 10      （将 VLAN 2 的端口权值设为 10）
Switch1(config-if)# spanning-tree vlan 3 port-priority 128     （将 VLAN 3 的端口权值设为 128）
Switch1(config-if)# spanning-tree vlan 4 port-priority 128     （将 VLAN 4 的端口权值设为 128）
Switch1(config-if)# spanning-tree vlan 5 port-priority 128     （将 VLAN 5 的端口权值设为 128）
Switch1(config-if)#exit
Switch1(config)#interface f0/24                                （进入端口 24 配置模式，Trank2）
Switch1(config-if)# spanning-tree vlan 1 port-priority 128     （将 VLAN 1 的端口权值设为 128）
Switch1(config-if)# spanning-tree vlan 2 port-priority 128     （将 VLAN 2 的端口权值设为 128）
Switch1(config-if)# spanning-tree vlan 3 port-priority 10      （将 VLAN 3 的端口权值设为 10）
Switch1(config-if)# spanning-tree vlan 4 port-priority 10      （将 VLAN 4 的端口权值设为 10）
Switch1(config-if)# spanning-tree vlan 5 port-priority 10      （将 VLAN 5 的端口权值设为 10）
Switch1(config-if)#end
Switch1#copy running-config startup-config                     （保存配置文件）
```

这样，由于分别设置了不同 Trunk 上不同 VLAN 的权值，而默认情况下的权值为 128，STP 协议就可以根据权值的大小来使 Trunk1 发送和接收 VLAN1-2 的数据，Trunk2 发送和接收 VLAN3-5 的数据，来实现负载均衡的目的。

2．配置 STP 路径值的负载均衡

也可以通过配置 STP 路径值来实现负载均衡，如图 8-17 所示。Trunk1 走 VLAN1-2 的数据，Trunk2 走 VLAN3-5 的数据。

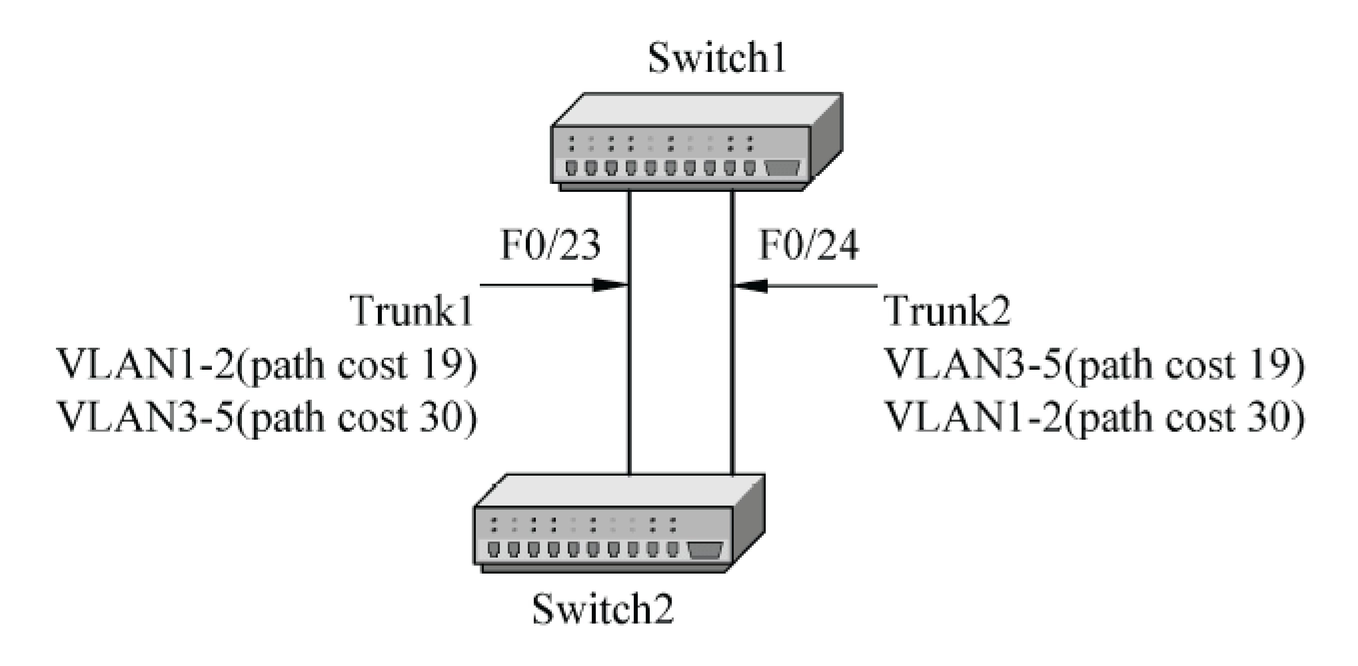
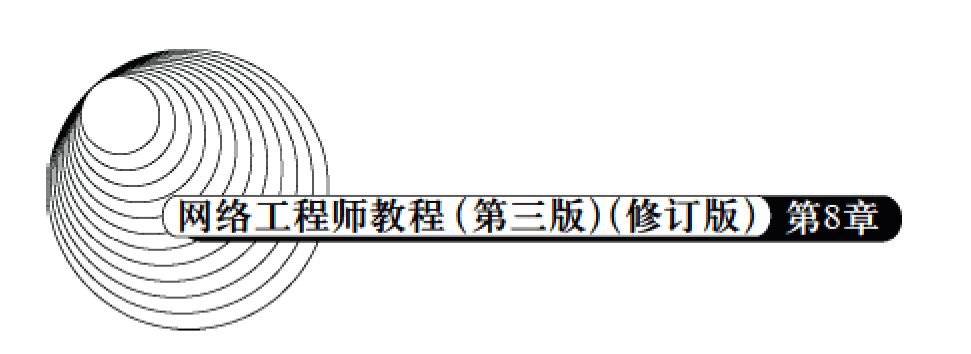


图 8-17　STP 路径值的负载均衡



其中，VTP 及 VLAN Trunk 的配置和上面相同，在此不再列出。只说明在配置好 VTP 协议和 VLAN Trunk 端口后在服务器（Switch1）上如何配置 STP 路径值。

```
Switch1#config Terminal
Switch1(config)#interface f0/23                      （进入端口 23 配置模式，配置 Trunk1）
Switch1(config-if)#spanning-tree vlan 1 cost 19      （设置 VLAN3 生成树路径值为 19）
Switch1(config-if)#spanning-tree vlan 2 cost 19      （设置 VLAN4 生成树路径值为 19）
Switch1(config-if)#spanning-tree vlan 3 cost 30      （设置 VLAN3 生成树路径值为 30）
Switch1(config-if)#spanning-tree vlan 4 cost 30      （设置 VLAN4 生成树路径值为 30）
Switch1(config-if)#spanning-tree vlan 5 cost 30      （设置 VLAN5 生成树路径值为 30）
Switch1(config-if)#exit
Switch1(config)#interface f0/24                      （进入端口 24 配置模式，配置 Trunk1）
Switch1(config-if)#spanning-tree vlan 1 cost 30      （设置 VLAN3 生成树路径值为 30）
Switch1(config-if)#spanning-tree vlan 2 cost 30      （设置 VLAN4 生成树路径值为 30）
Switch1(config-if)#spanning-tree vlan 3 cost 19      （设置 VLAN3 生成树路径值为 19）
Switch1(config-if)#spanning-tree vlan 4 cost 19      （设置 VLAN4 生成树路径值为 19）
Switch1(config-if)#spanning-tree vlan 5 cost 19      （设置 VLAN5 生成树路径值为 19）
Switch1(config-if)#end
Switch1#
```

这样，将希望阻断的 VLAN 生成树路径设大，STP 协议就会阻断该 VLAN 从该 Trunk 上通过，从而可以把负载均衡到多个 Trunk 端口上。

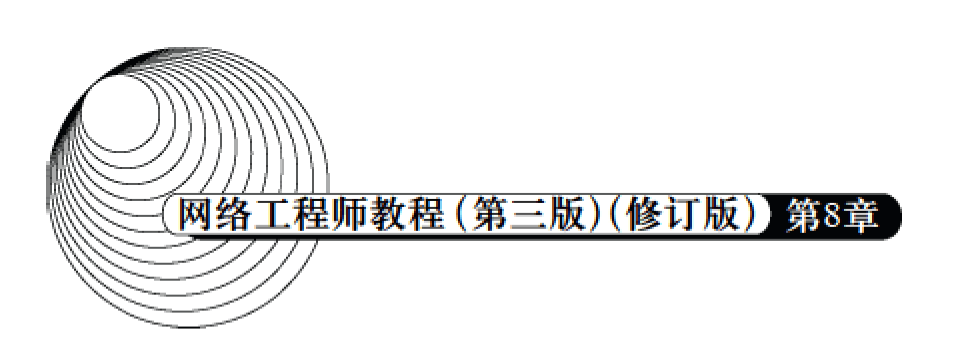
8.3 路由器的配置

现在市场上路由器也是种类繁多、型号各异，但 Cisco 的中高端路由器仍然占市场主流，也具有一定的代表性。本节就结合 Cisco 路由器来讲解有关路由器配置的相关技术和知识。

8.3.1 路由器概述

路由器（Router）是一种典型的网络层设备，在 OSI 参考模型中被称为中介系统，完成网络层中继或第三层中继的任务。路由器负责在两个局域网的网络层间接传输数据分组，并确定网络上数据传送的最佳路径。也因为它们运行 IP 协议基于第三层信息来为分组选择路由（如图 8-18 所示），所以路由器已经成为 Internet 的骨干。

路由器是用于连接多个逻辑上分开的网络，所谓逻辑网络，是代表一个单独的网络或者一个子网。当数据从一个子网传输到另一个子网时，可通过路由器来完成。因此，路由器具有判断网络地址和选择路径的功能，它能在多网络互联环境中建立灵活的连接，可用完全不同的数据分组和介质访问方法连接各种子网。它不关心各子网使用的硬件设备，但要求运行与网络层



在特权模式下，用户可以发出丰富的命令，以便更好地控制和使用路由器；在配置模式下，用户可以创建和更改路由器的配置，对路由器的管理和配置主要工作在配置模式下。

其中，配置模式又分为全局配置模式和接口配置模式、路由协议配置模式、线路配置模式等子模式。在不同的工作模式下，路由器有不同的命令提示状态。

- Router>。路由器处于用户执行模式命令状态，这时用户可以看路由器的连接状态，访问其他网络和主机，但不能看到和更改路由器的设置内容。
- Router# 。在 Router>提示符下输入 enable，路由器进入特权命令状态 Router#，这时不但可以执行所有的用户命令，还可以看到和更改路由器的设置内容。
- Router(config)#。在 Router#提示符下输入 configure terminal，出现提示符 Router (config)#，此时路由器处于全局设置状态，这时可以设置路由器的全局参数。
- Router(config-if)#，router(config-line)#，router(config-router)#，…。路由器处于局部设置状态，这时可以设置路由器某个局部的参数。
- >。路由器处于 RXBOOT 状态，在开机后 60s 内按 Ctrl+break 组合键可进入此状态，这时路由器不能完成正常的功能，只能进行软件升级和手工引导。或者进行路由器口令恢复时要进入该状态。
- 设置对话状态。这是一台新路由器开机时自动进入的状态，在特权命令状态使用 setup 命令也可进入此状态，这时可通过对话方式对路由器进行设置。

2. 路由器的基本配置

配置 enable 口令、enable 密码和主机名，在路由器中同样可以配置使能口令（enable password）和使能密码（enable secret），一般情况下只需配置一个就可以，当两者同时配置时，后者生效。这两者的区别是使能口令以明文显示而使能密码以密文形式显示。主机名及路由器口令的设置和上节对交换机配置的主机名及口令相同，这里不再复述。

配置路由器以太网接口，路由器一般提供一个或多个以太网接口槽，每个槽上会有一个以上以太网接口。以太网接口也因此而命名为{Ethernet 槽位/端口}或{FastEthernet 槽位/端口}，例如 FastEthernet0/0、FastEthernet1/1，也可缩写为 F0/0、F1/1。

以 Cisco2600 系列路由器为例，电缆连接如图 8-19 所示，连接好仿真终端到路由器的 Console 电缆线，就可以对路由器进行初始的配置工作。配置以太网接口如下。

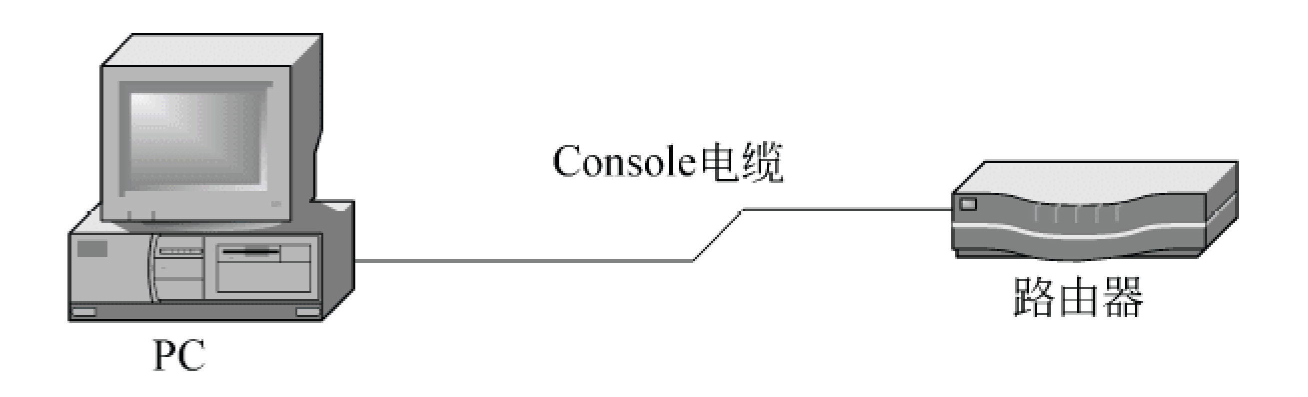
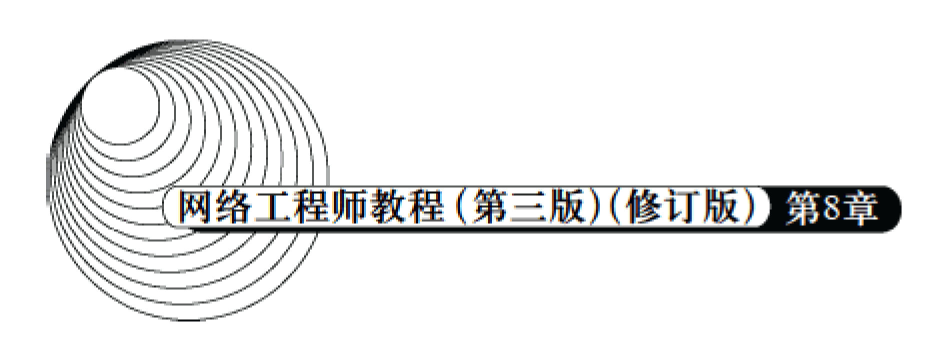


图 8-19　仿真终端与路由器的连接



在 line1-8 线路上设置 transport input all 来指明所有的协议可被用于连接到指定的路由器线路。根据以上要求配置好的终端服务器清单如下。

```
Current configuration
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Term_Server                      （主机名）
!
enable secret 5 $1$iMKo$htleQKTbMW4h1RbNXJiF9.
enable password 2600
!
ip subnet-zero
!
ip host router 2001 10.1.1.1              （主机表）
ip host router 2002 10.1.1.1
!
interface loopback0                       （回送接口）
 ip addr 10.1.1.1 255.255.255.255
interface FastEthernet0/0
 no ip address
!
interface FastEthernet0/1
 no ip address
!
ip classless
ip http server
ip pim bidir-enable
!
line con 0
line 1 8
 no exec
 transport input all
line aux 0
line vty 0 4
```

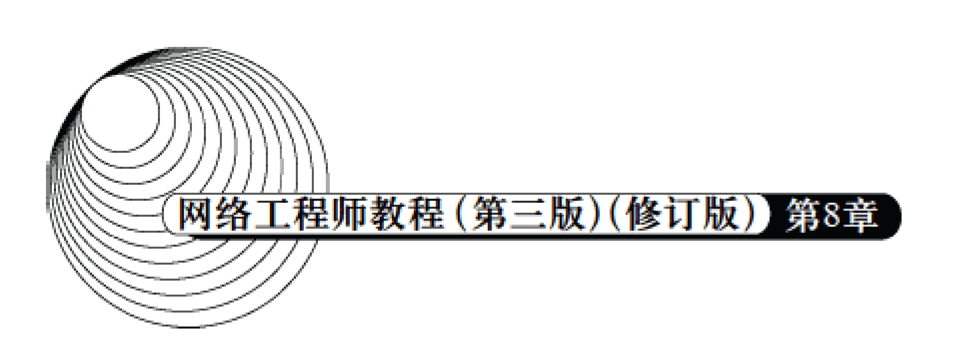
```
 password user11
 login
!
end
```

下面来看如何通过终端服务器访问其他路由器以及如何在多个路由器会话间切换。通过下面的配置清单来设置另外两个路由器的主机名。

```
Term_Server#
Term_Server#router1                          （访问主机表中的 router1 路由器）
Trying router1 (10.1.1.1, 2001)...Open
Router>enable
Router#config t
Enter configuration commands, one per line.   End with CNTL/Z.
Router(config)#hostname router1              （设置路由器 1 的主机名）
Router1(config)#end
Router1#
(会话切换命令 Ctrl+Shift+6 后接 x，即先同时按下 Ctrl+Shift+6 组合键，松开后再按下 x 键)
Term_Server#
Term_Server#router2
Trying router2 (10.1.1.1, 2002)...Open
Router>enable
Router#config t
Enter configuration commands, one per line.   End with CNTL/Z.
Router(config)#hostname router2
Router2(config)#end
Router2#
（会话切换命令 Ctrl+Shift+6 后接 x)
Term_Server#show sessions                    （查看终端服务器的会话）
Conn      Host       Address           Byte    Idle     Conn Name
  1       router1    10.1.1.1           0       0       router1
  2       router2    10.1.1.1           0       0       router2
Term_Server#disconnect 2                     （断开会话 2）
Term_Server#show line 1                      （查看线路 1 的状态）
Term_Server#clear line 2                     （清除线路 2）
```

4．配置静态路由

通过配置静态路由，用户可以人为地指定对某一网络访问时所要经过的路径，在网络结构



比较简单，且一般到达某一网络所经过的路径唯一的情况下采用静态路由。通过以下实例来让大家掌握静态路由的设置、查看路由表，理解路由原理及概念。

1）IPv4 静态路由设置

如图 8-21 所示设计拓扑结构， 3 台路由器分别命名为 R1、R2 和 R3，所使用的接口和相应的 IP 地址分配如图中所示，其中“/24”表示子网掩码为 24 位，即 255.255.255.0。配置中所用到的终端服务器不在图中标出。

要在路由器中配置静态路由以实现路由器 R2 到 R3 在 IP 层的连通性，也就是要求从 R2 可以 ping 通 R3，从 R3 可以 ping 通 R1。

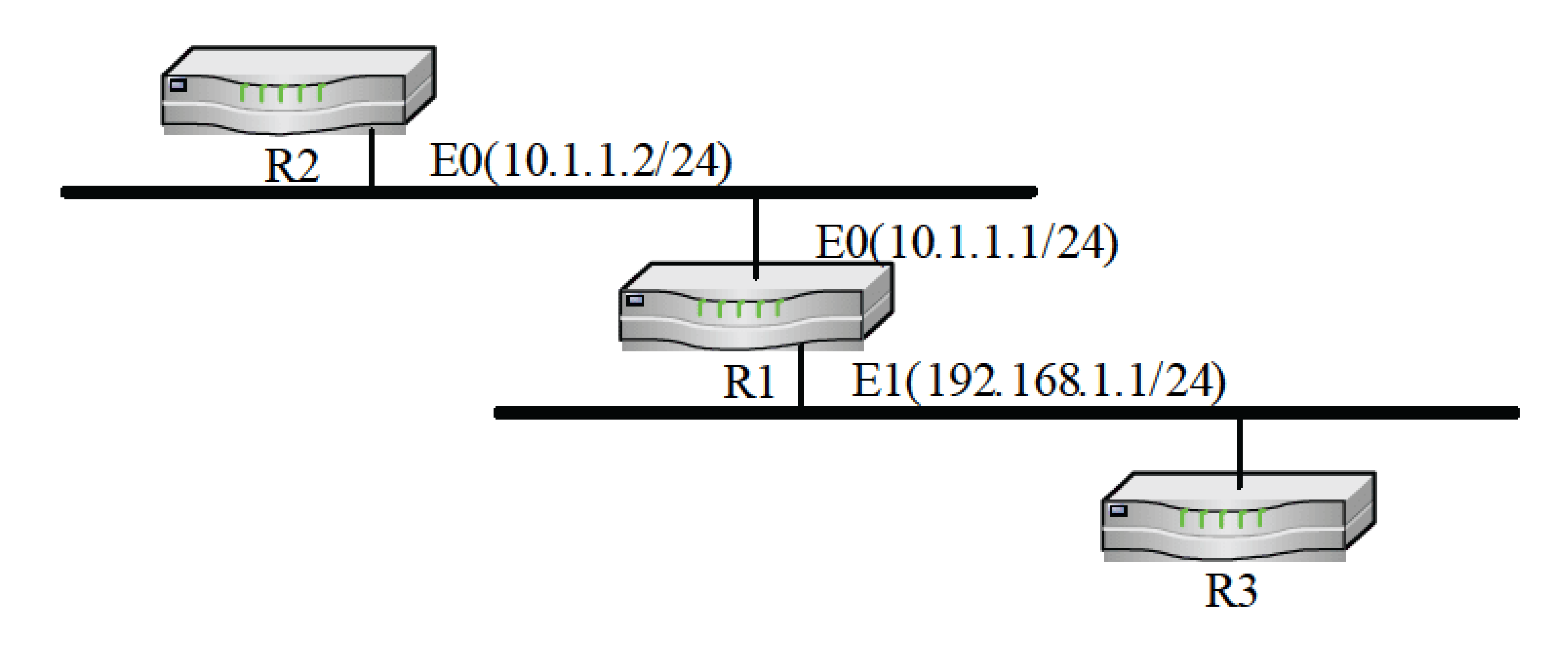


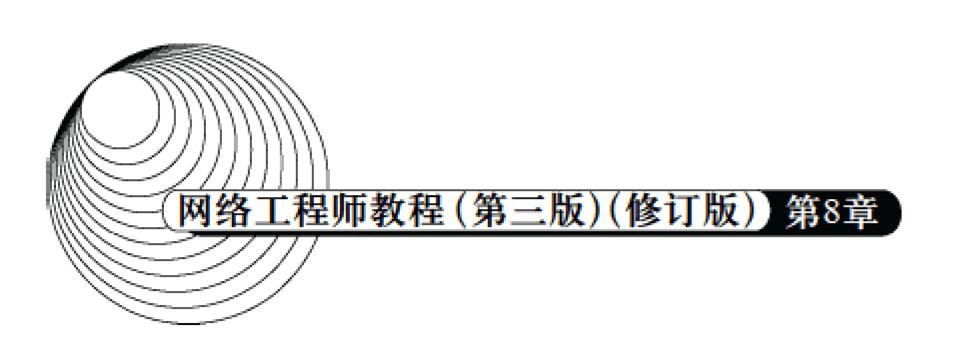
图 8-21　静态路由实例图

首先根据拓扑结构图配置各路由器的以太网接口，配置方法在前面已经讲过，这里不再重复。从接口的配置工作完成之后开始讲解静态路由的配置。

配置好以太网接口后来测试路由器间基本的连通性。首先从 R1 路由器 ping 路由器 R2 和 R3。

```
R1#ping 10.1.1.2                    （ping R2，结果连通）
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, time out is 2 seconds:
!!!!!!
Success rate is 100 percent(5/5), round-trip min/avg/nax=4/4/4 ms
R1#ping 192.168.1.3                 （ping R3，结果连通）
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, time out is 2 seconds:
!!!!!!
Success rate is 100 percent(5/5), round-trip min/avg/nax=4/4/4 ms
```

从 R2 路由器 ping 路由器 R1 的 E1 接口。



ip icmp 命令来监视 IP 包和 ICMP 包的传输情况而知。

```
R2#ping 192.168.1.3                （ping R3 的 E0 接口，结果不连通）
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, time out is 2 seconds:
!!!!!!
Success rate is 0 percent(0/5)
```

此时需要在 R3 上加入发往 R2 网段数据包的路由信息。

```
R3#config t
R3(config)#ip route 10.1.1.0 255.255.255.0 192.168.1.1          （加入静态路由）
R3(config)#end
```

这样一来，就可以实现路由器 R2 到 R3 之间的连通性，互相可以 ping 通对方接口的 IP 地址。应该注意的是，在有些路由器上默认情况是不启动 IP 路由的，这时可以用 ip routing 和 no ip routing 来启动和关闭 ip 路由。

2）IPv6 静态路由设置

网络拓扑结构如图 8-22 所示，两台路由器分别命名为 R1 和 R2，所使用的接口和相应的 IP 地址分配如图中所示。

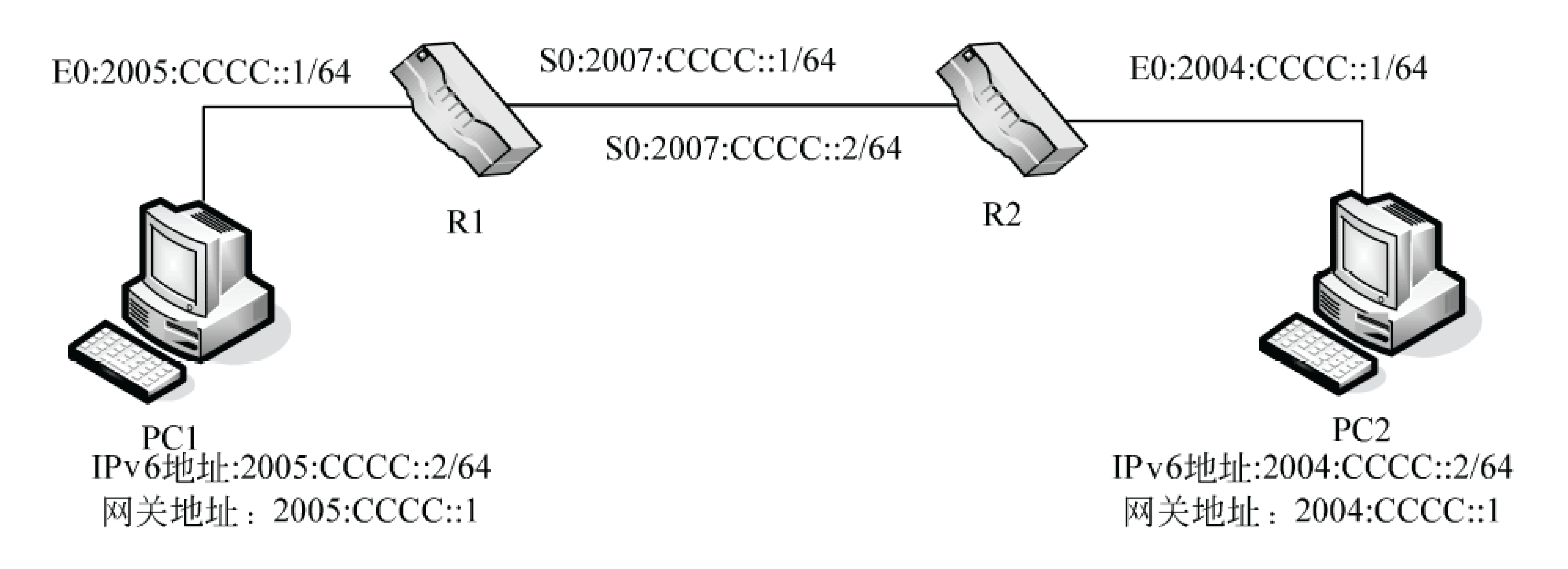
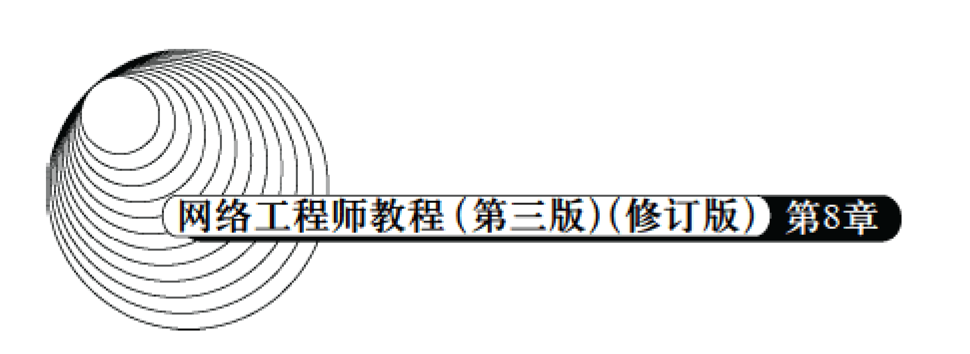


图 8-22 IPv6 静态路由实例图

要在路由器中配置 IPv6 静态路由以实现 PC1 到 PC2 在 IP 层的连通性，也就是要求从 PC1 可以 ping 通 PC2。

R1 相关配置如下。

```
Router#
Router# configure terminal
```




```
    2005:AAAA::1, subnet is 2005:AAAA::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
    FF02::1:FF9B:2201
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
Serial0/2/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2E0:B0FF:FEC9:1701
  No Virtual link-local address(es):
  Global unicast address(es):
    2007:CCCC::1, subnet is 2007:CCCC::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
    FF02::1:FFC9:1701
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
Vlan1 is administratively down, line protocol is down
  Internet protocol processing disabled
```

在 pc1 上配置 IPv6 地址 2005: CCCC::2/64，在 pc2 上配置 IPv6 地址 2004: CCCC::2/64，然后在 pc1 上通过 ping 命令测试与 pc2 的连通性，结果如下。

```
C:\>ping 2005: CCCC::2
Pinging 2005: CCCC::2 with 32 bytes of data:
Reply from 2005: CCCC::2: bytes=32 time=109ms TTL=126
Reply from 2005: CCCC::2: bytes=32 time=78ms TTL=126
Reply from 2005: CCCC::2: bytes=32 time=94ms TTL=126
Reply from 2005: CCCC::2: bytes=32 time=94ms TTL=126
Ping statistics for 2005: CCCC::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 78ms, Maximum = 109ms, Average = 93ms
```

8.4 配置路由协议

本节主要讲述对路由协议的配置。IP 路由选择协议用有效的、无循环的路由信息填充路由表，从而为数据包在网络之间传递提供了可靠的路径信息。路由选择协议又分为距离矢量、链路状态和平衡混合三种。

距离矢量（Distance Vector）路由协议计算网络中所有链路的矢量和距离并以此为依据确认最佳路径。使用距离矢量路由协议的路由器定期向其相邻的路由器发送全部或部分路由表。典型的距离矢量路由协议有 RIP 和 IGRP。

链路状态（Link State）路由协议使用为每个路由器创建的拓扑数据库来创建路由表，每个路由器通过此数据库建立一个整个网络的拓扑图。在拓扑图的基础上通过相应的路由算法计算出通往各目标网段的最佳路径，并最终形成路由表。典型的链路状态路由协议是 OSPE（Open Shortest Path First，开放最短路径优先）路由协议。

平衡混合（Balanced Hybrid）路由协议结合了链路状态和距离矢量两种协议的优点，此类协议的代表是 EIGRP，即增强型内部网关协议。

下面将分别讨论如何在路由器中配置这些动态路由协议。

8.4.1 配置 RIP 协议

RIP（路由选择信息协议）是距离矢量路由选择协议的一种。路由器收集所有可到达目的地的不同路径，并且保存有关到达每个目的地的最少站点数的路径信息，除到达目的地的最佳路径外，任何其他信息均予以丢弃。同时，路由器也把所收集的路由信息用 RIP 协议通知相邻的其他路由器。这样，正确的路由信息逐渐扩散到了全网。

RIP 使用非常广泛，它简单、可靠，便于配置。RIP 版本 2 还支持无类域间路由（Classless Inter-Domain Routing，CIDR）和可变长子网掩码（Variable Length Subnetwork Mask，VLSM）

及不连续的子网，并且使用组播地址发送路由信息。但是，RIP 只适用于小型的同构网络，因为它允许的最大跳数为 15，任何超过 15 个站点的目的地均被标记为不可达。RIP 每隔 30s 广播一次路由信息。

相关的命令如表 8-3 所示。

表 8-3　RIP 相关命令

命　　令	功　　能	命　　令	功　　能
router rip	指定使用 RIP 协议	show ip route	查看路由表信息
version {1|2}	指定 RIP 版本	show ip route rip	查看 RIP 协议路由信息
network *network*	指定与该路由器相连的网络		

假设有图 8-23 所示的网络拓扑结构，试通过配置 RIP 协议使全网连通。在配置之前先按照拓扑结构连接好网络设备，其中串口之间需要用 DTE（数据终端设备）和 DCE（数据通信设备）电缆对接，或者用 DCE 转 DTE 电缆连接。

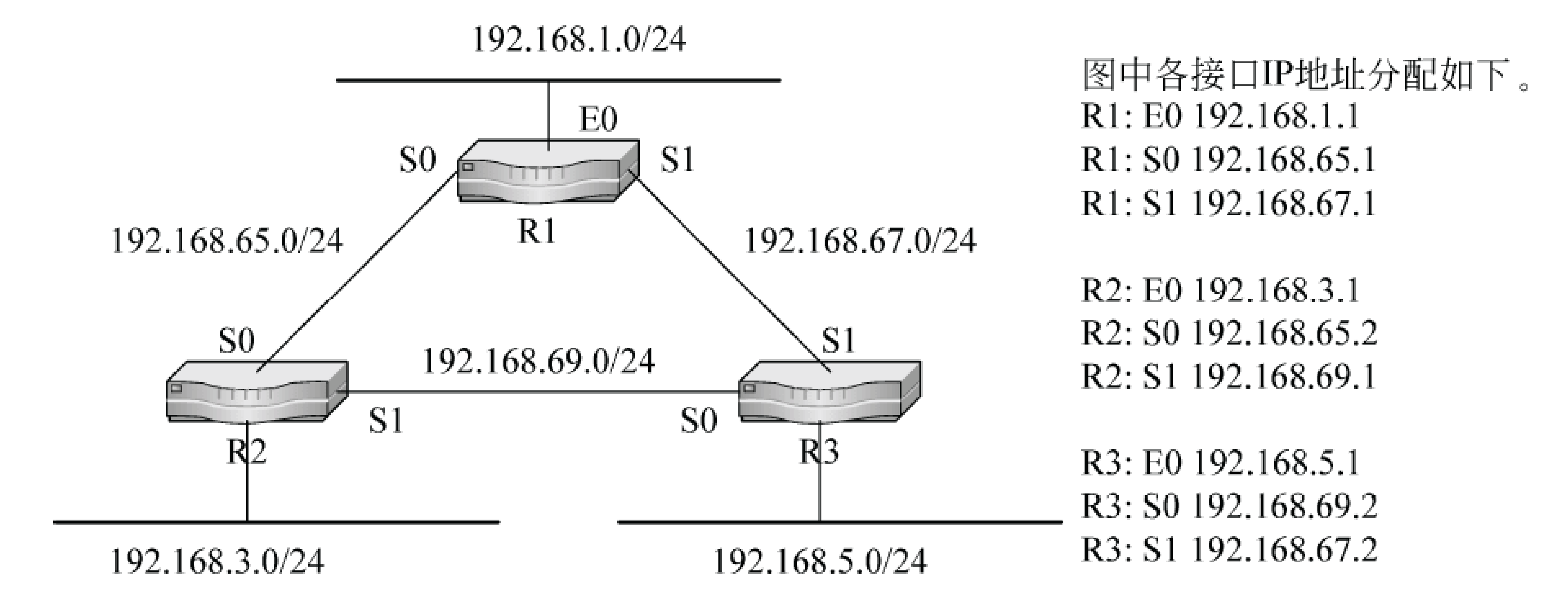


图 8-23　RIP 协议配置拓扑图

首先根据图中要求配置各路由器的各接口地址。

```
R1#config t
R1 (config)#no logging console
R1 (config)#interface fastethernet0/1
R1 (config-if)#ip address 192.168.1.1 255.255.255.0
R1 (config-if)#no shutdown
R1 (config-if)#exit
R1 (config)#interface serial 0
R1 (config-if)#ip address 192.168.65.1 255.255.255.0
R1 (config-if)#no shutdown
```

```
R1 (config-if)#exit
R1 (config)#interface serial 1
R1 (config-if)#ip address 192.168.67.1 255.255.255.0
R1 (config-if)#no shutdown
```

在全局配置模式下使用 no logging console 配置命令，可以防止大量的端口状态变化信息和报警信息对配置过程的影响。为了查明串行接口所连接的电缆类型，从而正确配置串行接口，可以使用 show controllers serial 命令来查看相应的控制器。注意，在配置端口时使用 no shutdown 命令，因为默认情况下各物理接口是处于关闭状态，配置完成需要对端口进行激活。

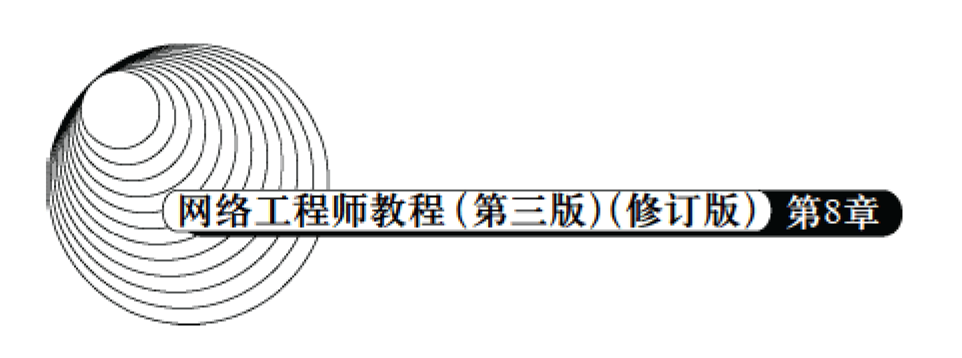
类似配置 R1 各接口地址的方法可以配置好路由器 R2 和 R3 的各接口地址。此时路由表中只有和路由器直接相连的各网段的路由信息，即每个路由器只可以 ping 通和它直接相连的路由器的接口。此时可以用 show ip route 命令查看路由表信息。

```
R1#show ip route
Codes: C – connected, S – static, I – IGRP, R – RIP, M – mobile, B – BGP
       D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
       N1 – OSPF NSSA external type 1, N2 – OSPF NSSA external type 2
       E1 – OSPE external type 1, E2 – OSPF external type 2, E – EGP
       I – IS-IS, L1 – IS-IS level-1, L2 – IS-IS level-2, ia – IS-IS inter area
       * - candidate default U – per-user static route, o - ODR
       P – periodic downloaded static route
Gateway of last resort is not set
     192.168.0.0/24 is subnetted, 3 subnets
C         192.168.1.0 is directly connected, Ethernet0
C         192.168.65.0 is directly connected, Serial0
C         192.168.67.0 is directly connected, Serial1
```

配置完接口地址后就可以进行 RIP 协议配置，RIP 协议配置非常简单。首先使用 ip routing 允许路由选择协议，在有些路由器上默认情况是关闭的。用 router rip 命令进入 RIP 协议配置模式，然后使用 network 语句声明进入 RIP 进程的网络就可以了。

配置路由器 R1。

```
R1 (config)#ip routing
R1 (config)#router rip                      （进入 RIP 协议配置子模式）
R1 (config-router)#network 192.168.1.0      （声明网络 192.168.1.0/24）
R1 (config-router)#network 192.168.65.0
R1 (config-router)#network 192.168.67.0
R1 (config-router)#version 2                （设置 RIP 协议版本 2）
R1 (config-router)#exit
```



配置路由器 R2。

```
R1 (config)#ip routing
R1 (config)#router rip                      （进入 RIP 协议配置子模式）
R1 (config-router)#network 192.168.3.0      （声明网络 192.168.3.0/24）
R1 (config-router)#network 192.168.65.0
R1 (config-router)#network 192.168.69.0
R1 (config-router)#version 2                （设置 RIP 协议版本 2）
R1 (config-router)#exit
```

配置路由器 R3。

```
R1 (config)#ip routing
R1 (config)#router rip                      （进入 RIP 协议配置子模式）
R1 (config-router)#network 192.168.5.0      （声明网络 192.168.5.0/24）
R1 (config-router)#network 192.168.67.0
R1 (config-router)#network 192.168.69.0
R1 (config-router)#version 2                （设置 RIP 协议版本 2）
R1 (config-router)#exit
```

配置完 RIP 协议后，RIP 协议的路由器广播自己的路由信息到周边路由器，此时各路由器就可以学习到其他路由器的路由信息。此时再查看路由信息则有所不同，下面来查看 R3 上的路由表。

```
R3#show ip route
Codes: C – connected, S – static, I – IGRP, R – RIP, M – mobile, B – BGP
       D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
       N1 – OSPF NSSA external type 1, N2 – OSPF NSSA external type 2
       E1 – OSPE external type 1, E2 – OSPF external type 2, E – EGP
       I – IS-IS, L1 – IS-IS level-1, L2 – IS-IS level-2, ia – IS-IS inter area
       * - candidate default U – per-user static route, o - ODR
       P – periodic downloaded static route
Gateway of last resort is not set
     192.168.0.0/24 is subnetted, 6 subnets
C        192.168.1.0 is directly connected, Ethernet0
C        192.168.65.0 is directly connected, Serial0
C        192.168.67.0 is directly connected, Serial1
R        192.168.65.0 [120/1] via 192.168.67.1, 00:00:15, Serial
[120/1] via 192.168.69.1, 00:00:24, Seria0
R        192.168.1.0 [120/1] via 192.168.67.1, 00:00:15, Serial
R        192.168.3.0 [120/1] via 192.168.69.1, 00:00:24, Seria0
```

路由表中的项目解释如下。

R　　　192.168.3.0 [120/1] via 192.168.69.1, 00:00:24, Seria0

- R：表示此项路由是由 RIP 协议获取的，另外 C 代表直接相连的网段。
- 192.168.3.0：表示目标网段。
- [120/1]：120 表示 RIP 协议的管理距离默认为 120，1 是该路由的度量值，即跳数。
- via：经由的意思。
- 192.168.69.1：表示从当前路由器出发到达目标网的下一跳点的 IP 地址。
- 00:00:24：　　表示该条路由产生的时间。
- Seria0：表示该条路由使用的接口。

从路由表中可以看出多了三条 RIP 路由信息，这三条路由信息分别可以访问到网络 192.168.65.0/24、192.168.1.0/24 和 192.168.3.0/24，其中访问到 192.168.65.0/24 的路由有两条。之所以保存两条路由信息，是因为到达目的网段需要经过的跳数相同，都为 1。而访问另外两个网段也可以经过另外两个路由器来转发，但是因为那样要经过两跳，而 RIP 协议是选择跳数作为唯一度量路由选择的标准，所以它只将跳数最少的路径保留在路由表中，而其余的路径都被放弃。

另外，因为 RIP 版本 2 支持不连续子网和可变长子网掩码，所以各网段的 IP 地址可以是任何形式的合法 IP。通过以上对各路由器配置 RIP 协议可以达到全网连通的目的。

8.4.2　配置 IGRP 协议

内部网关路由协议（Interior Gateway Routing Protocol，IGRP）是一种动态距离向量路由协议，它是 Cisco 公司 20 世纪 80 年代中期设计的。使用组合用户配置尺度，包括延迟、带宽、可靠性和负载。它能够变通地处理不确定的、复杂的拓扑结构，不支持 VLSM 和不连续的子网。

默认情况下，IGRP 每 90s 发送一次路由更新广播，在 3 个更新周期内（即 270s）没有从路由表中的一个路由器接收到更新，则宣布路由不可访问。在 7 个更新周期即 630s 后，IOS 软件从路由表中清除路由。

相关命令如表 8-4 所示。

表 8-4　IGRP 相关命令

命　　令	功　　能
router igrp autonomous-system	指定使用 IGRP 协议
network network	指定与该路由器相连的网络
show ip route	查看路由表信息
show ip route igrp	查看 IGRP 协议路由信息

```
 ip address 192.168.5.1 255.255.255.0
 no keepalive
!
interface Serial0
 ip address 192.168.69.2 255.255.0.0
 bankwidth 500
!
interface Serial1
 ip address 192.168.67.2 255.255.0.0
 bankwidth 64
!
router igrp 100
 network 192.168.5.0
 network 192.168.69.0
 network 192.168.67.0
!
```

这里需要指出的是，以太网接口中的 no keepalive 能使此接口不监测 keepalive（存活）信号，从而在不连接任何设备的情况下可以激活此接口。这样只是为我们配置和监测路由协议而设，在实际网络环境中是不应该使用该命令的。使用 router igrp 100 创建 IGRP 路由进程，后面的 100 是自治系统号。三个路由器的自治系统号必须相同，否则彼此的路由信息将不被互相传递和学习。

配置工作完成后可以分别查看三个路由器的路由表，看看和 RIP 协议配置完成所产生的路由表有何区别。

```
R1#show ip route igrp
     192.168.0.0/24 is subnetted, 6 subnets
I          192.168.69.0 [100/160250] via 192.168.67.2, 00:00:15, Serial
[100/160250] via 192.168.65.2, 00:00:24, Seria0
I          192.168.3.0 [100/22100] via 192.168.65.2, 00:00:15, Seria0
I          192.168.5.0 [100/22100] via 192.168.67.2, 00:00:15, Seria1
R2#show ip route igrp
     192.168.0.0/24 is subnetted, 6 subnets
I          192.168.67.0 [100/24000] via 192.168.65.1, 00:00:21, Seria0
I          192.168.1.0 [100/22100] via 192.168.65.1, 00:00:21, Seria0
I          192.168.5.0 [100/24100] via 192.168.65.1, 00:00:21, Seria0
R3#show ip route igrp
```

```
     192.168.0.0/24 is subnetted, 6 subnets
I          192.168.65.0 [100/24000] via 192.168.67.1, 00:00:21, Serial
I          192.168.1.0 [100/22100] via 192.168.67.1, 00:00:21, Serial
I          192.168.3.0 [100/24100] via 192.168.67.1, 00:00:21, Serial
```

从路由表可以看出，R2 访问 R3 和 R3 访问 R2 的路由都要经过路由器 R1 转发，这是因为 IGRP 协议计算度量值时是考虑网络带宽的，它根据所计算出的度量值，选择度量值最小的路径保留在路由表中，其中 100 是 IGRP 协议的管理距离。

8.4.3 配置 OSPF 协议

开放最短路径优先协议是重要的路由选择协议。它是一种链路状态路由选择协议，是由 Internet 工程任务组开发的内部网关路由协议，用于在单一自治系统内决策路由。

链路是路由器接口的另一种说法，因此 OSPF 也称为接口状态路由协议。OSPF 通过路由器之间通告网络接口的状态来建立链路状态数据库，生成最短路径树，每个 OSPF 路由器使用这些最短路径构造路由表。下面分别介绍 OSPF 协议的相关要点。

（1）自治系统。自治系统包括一个单独管理实体下所控制的一组路由器，OSPF 是内部网关路由协议，工作于自治系统内部。

（2）链路状态。所谓链路状态，是指路由器接口的状态，如 Up、Down、IP 地址、网络类型以及路由器和它邻接路由器间的关系。链路状态信息通过链路状态通告（Link State Advertisement，LSA）扩散到网上每台路由器，每台路由器根据 LSA 信息建立一个关于网络的拓扑数据库。

（3）最短路径优先算法。OSPF 协议使用最短路径优先算法，利用从 LSA 通告得来的信息计算每一个目标网络的最短路径，以自身为根生成一棵树，包含了到达每个目的网络的完整路径。

（4）路由标识。OSPF 的路由标识是一个 32 位的数字，它在自治系统中被用来唯一识别路由器。默认地使用最高回送地址，若回送地址没有被配置，则使用物理接口上最高的 IP 地址作为路由器标识。

（5）邻居和邻接。OSPF 在相邻路由器间建立邻接关系，使它们交换路由信息。邻居是指共享同一网络的路由器，并使用 Hello 包来建立和维护邻居路由器间的关系。

（6）区域。在 OSPF 网络中使用区域（Area）来为自治系统分段。OSPF 是一种层次化的路由选择协议，区域 0 是一个 OSPF 网络中必须具有的区域，也称为主干区域，其他所有区域要求通过区域 0 互连到一起。

相关命令及说明如表 8-5 所示。

```
R1#show running-config
!
interface Serial0
 ip address 192.200.10.1 255.255.255.252
!
interface Ethernet0
ip address 10.20.10.1 255.255.255.255
!
router eigrp 200
network 192.200.10.0 0.0.0.3
network 10.20.10.0 0.0.0.255
no auto-summary
!
R2#show running-config
!
interface Serial0
 ip address 192.200.10.2 255.255.255.252
!
interface Ethernet0
ip address 201.10.10.1 255.255.255.255
!
router eigrp 200
network 192.200.10.0 0.0.0.3
network 201.10.10.0
no auto-summary
!
```

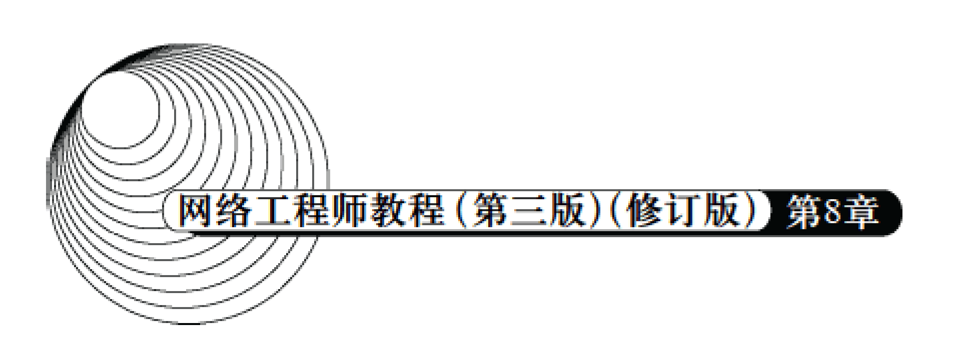
配置完成后可以使用 debug 命令和 show 命令来调试和检查配置结果，查看路由信息是否已经学习到，查看和调试方法和其他协议类似，这里不再重复。

8.5 配置广域网接入

要将网络与其他远程网络连接起来，有时就要用到广域网（WAN）接入服务。WAN 提供了与 LAN 不同的连接方法和布线标准。广域网中路由器和交换机连接方式多样化，主要有串行连接、ISDN BRI 连接和 DSL 连接等。本节结合具体接入实例来学习广域网接入的配置方法和技巧。

8.5.1 配置 ISDN

综合业务数字网（Integrated Service Digital Network，ISDN）是电话网络数字化的结果，



由数字电话和数据传输服务两部分组成。可以在 ISDN 上传输声音、数据和视频等多种信息。ISDN 组件包括终端、终端适配器、网络终端设备、线路终端设备和交换终端设备等。

ISDN 提供两种类型的访问接口，即基本速率接口（Basic Rate Interface，BRI）和主要速率接口（Primary Rate Interface，PRI）。ISDN BRI 提供两个 B 信道和一个 D 信道（2B+D）。ISDN 的 B 信道为承载信道，其速率为 64Kbps，用于传输用户数据；D 信道速率为 16Kbps，主要用于传输控制信息。PRI 提供 30 个 B 信道和 1 个 D 信道（30B+D），其 B 信道和 D 信道的速率均为 64Kbps。

下面通过一个具体的实例来学习两台路由器通过 ISDN 线路进行连接时的最基本配置。图 8-27 所示路由器 R1 和 R2 各连接 1 条 ISDN BRI 线路，路由器的 BRI 接口通过 NT1 连接到 ISDN 上。各路由器 BRI 接口的 IP 地址和所连接的 ISDN 号如图中所标。通过对两路由器的配置达到 R1 和 R2 互通的目的。

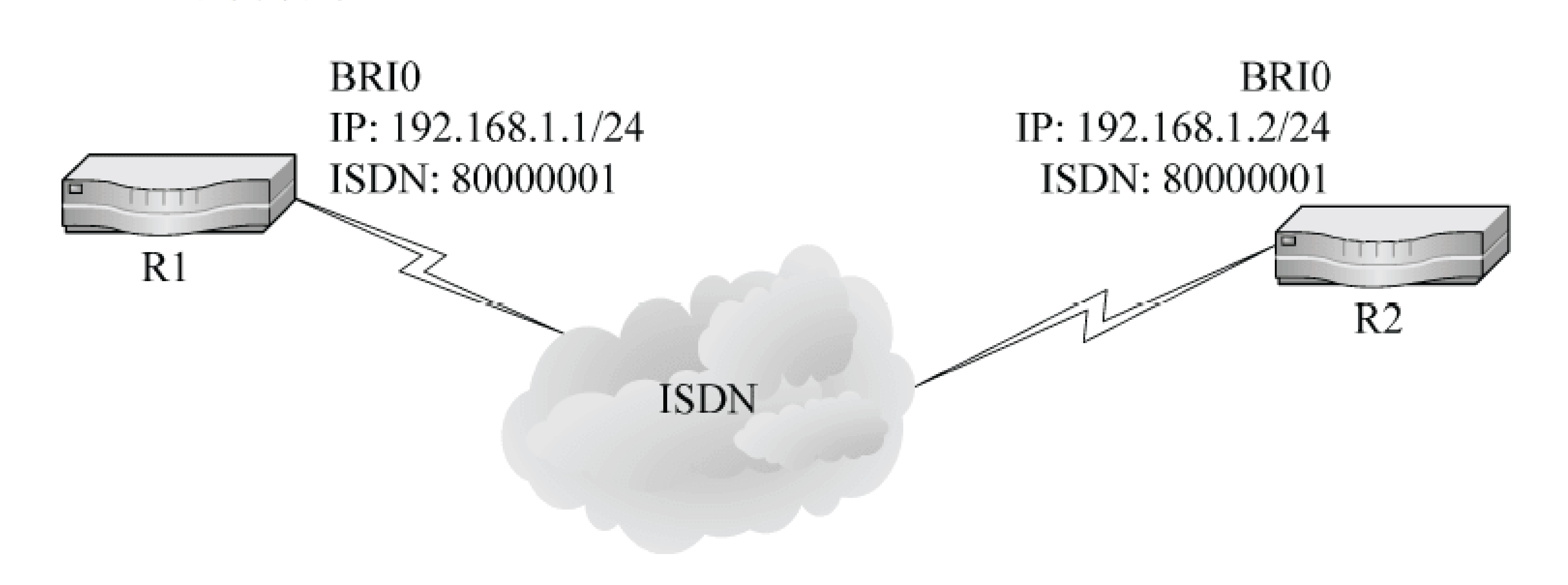


图 8-27　配置实例

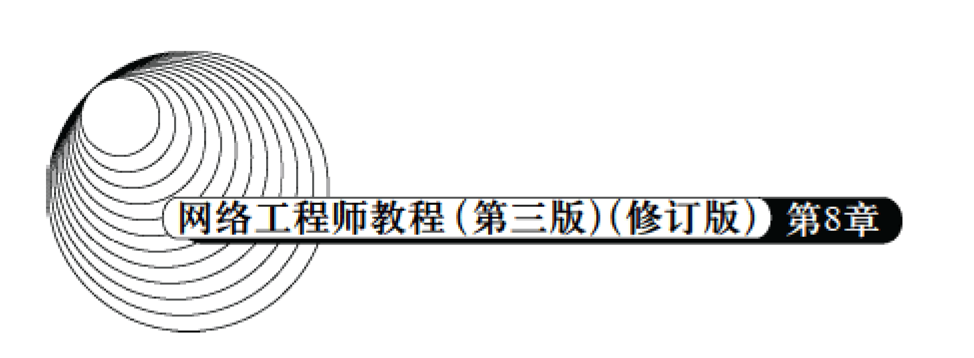
相关命令及说明如表 8-6 所示。

表 8-6　ISDN 相关配置命令

命　　令	功　　能
isdn switch-type *switch-type*[1]	设置 ISDN 交换类型
interface bri 0	接口 BRI 设置
encapsulation ppp	设置 PPP 封装
dialer map protocol *next-hop-address* [name *hostname*] [broadcast] [*dial-string*]	设置协议地址与电话号码的映射
ppp multilink	启动 PPP 多连接
dialer load-threshold *load*	设置启动另一个 B 通道的阈值
show isdn {active | history | memory | services | status [dsl | interface-type number] | timers}	显示 ISDN 有关信息

注：交换机类型 *switch-type* 可以用命令 isdn switch-type ?查得。

连接好线路之后，就可以进行配置工作。




```
R2(config-if)#dialer string 80000001          （设置拨号串，R1 的 ISDN 号码）
R2(config-if)#dialer-group 1                  （设置拨号组号为 1，把 BRI 0 接口与拨
                                                号列表 1 相关联）
R2(config-if)#no shutdown                     （激活接口）
R2(config-if)#exit
R2(config)#dialer-list 1 protocol ip permit   （设置拨号列表 1）
R2(config)#end
R2#
```

配置完成后，可以使用 debug 和 ping 命令来调试配置结果。其中阴影部分表示了拨号的过程。Ping 命令引发多次拨号行为，最后报告 BRI0：1 接口的线路协议已经激活。

```
R1(config)#logging console                    （在终端上显示监测信息）
R1(config)#exit
R1#debug dialer                               （监测 dialer 信息）
Dial on demand events debugging is on
R1#ping 192.168.1.2
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

02:11:13: BR0 DDR: Dialing cause ip(s=192.168.1.1, d=192.168.1.2)
02:11:13: BR0 DDR: Attempting to dial 80000002
02:11:15: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
02:11:15: %ISDN-6-CONNECT: Interface BRI0:1, is now connected to 80000002
.!!!
Success rate is 60 percent(3/5), round-trip min/avg/max = 36/38/40 ms
02:11:17: BR0:1 DDR: dialer protocol up
02:11:18: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
R1#undebug all                                （关闭所有调试信息）
```

还可以用 show isdn status 命令查看 ISDN 状态，用 show dialer 命令显示当前的拨号及其配置等信息，这里不再一一列出。

8.5.2 配置 PPP 和 DDR

点对点协议是作为在点对点链路上进行 IP 通信的封装协议而被开发出来的。PPP 定义了 IP 地址的分配和管理、异步和面向位的同步封装、网络协议复用、链路配置、链路质量测试和错误检测等标准，以及网络层地址协议和数据压缩协议等协议标准。PPP 通过可扩展的链路协议和网络控制协议（NCP）来实现上述功能。

（4）设置多链路。使用 dialer load-threshold 命令来配合 ppp multilink，设置数值为 128，告诉路由器在负载达到第 1 个 B 信道带宽的 50%以上时启用第 2 个 B 信道。这个数值 *N* 的取值范围是 0～255，表示当实际负载占到第一个 B 信道的 *N*/255%时启动第 2 个 B 信道。当设置为 1 时，表示不论负载多大同时启动两个 B 信道。

```
dialer load-threshold 128
ppp multilink
```

（5）no cdp enable 表示禁止通过此接口传递 CDP 控制数据，通常在拨号线路上要禁用 CDP。

（6）ppp authentication chap 表示设置 PPP 认证方式为 CHAP。

8.5.3　配置帧中继

帧中继是一种高性能的 WAN 协议，运行在 OSI 参考模型的物理层和数据链路层。它是一种数据包交换技术，是 X.25 的简化版本。它省略了 X.25 的一些强健功能，如提供窗口技术和数据重发技术，而是依靠高层协议提供纠错功能，这是因为帧中继工作在更好的 WAN 设备上，这些设备较之 X.25 的 WAN 设备具有更可靠的连接服务和更高的可靠性，它严格地对应于 OSI 参考模型的最低两层，而 X.25 还提供第三层的服务，所以帧中继比 X.25 具有更高的性能和更有效的传输效率。

帧中继广域网的设备分为 DTE 和 DCE，路由器作为 DTE 设备。

帧中继技术提供面向连接的数据链路层通信，在每对设备之间都存在一条定义好的通信链路，且该链路有一个链路识别码。这种服务通过帧中继虚电路实现，每个帧中继虚电路都以数据链路识别码（DLCI）标识自己。DLCI 的值一般由帧中继服务提供商指定。帧中继即支持 PVC 也支持 SVC。

相关命令及说明如表 8-8 所示。

表 8-8　帧中继相关配置命令

命　　令	功　　能
encapsulation frame-relay[ietf]	设置 Frame Relay 封装
frame-relay lmi-type {ansi | cisco | q933a}	设置 Frame Relay LMI 类型
interface interface-type interface-number subinterface- number [multipoint|point-to-point]	设置子接口
frame-relay map protocol protocol-address dlci [broadcast]	映射协议地址与 DLCI
frame-relay interface-dlci dlci [broadcast]	设置 FR DLCI 编号

1．配置帧中继交换机

帧中继的配置需要一个帧中继的环境，而现在普通的路由器就可以配置成一个帧中继交换机。假设有一个具有三个串行接口的路由器，通过下面的配置来实现全网状的帧中继环境。所谓全网状的帧中继环境，是指在这个帧中继环境中任何两个节点间都存在一条虚电路。参考图8-29 连接网络设备，图中是构造一个有三个节点的全网状帧中继环境，每个接口上的 DLCI 都标明在图上，虚线箭头表示两节点间的虚电路。下面给出路由器的配置清单。

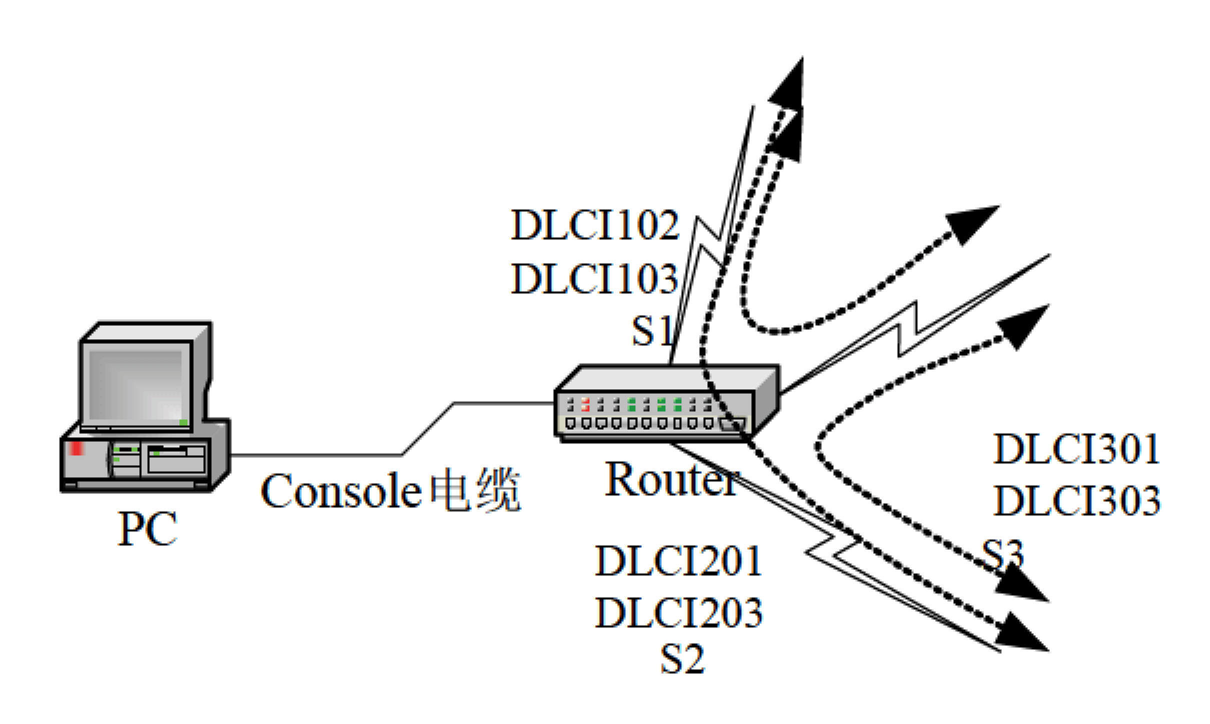


图 8-29　全网状帧中继环境示意图

```
Router#show run
!
interface serial1
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay lmi-type cisco
  frame-relay lmi-type dce
  frame-relay route 102 interface Serial2 201
  frame-relay route 103 interface Serial3 301
!
interface serial2
  no ip address
  encapsulation frame-relay
  clockrate 64000
  frame-relay lmi-type cisco
  frame-relay lmi-type dce
  frame-relay route 201 interface Serial1 102
```

```
R1(config)#interface s0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#encap frame-relay                    （该接口使用帧中继封装格式）
R1(config-if)#no shutdown
R1(config-if)#no frame-relay inverse-arp           （关闭帧中继逆向 ARP）
R1(config-if)#frame map ip 192.168.1.2 cisco
R1(config-if)#end
R1#
```

路由器 R2：

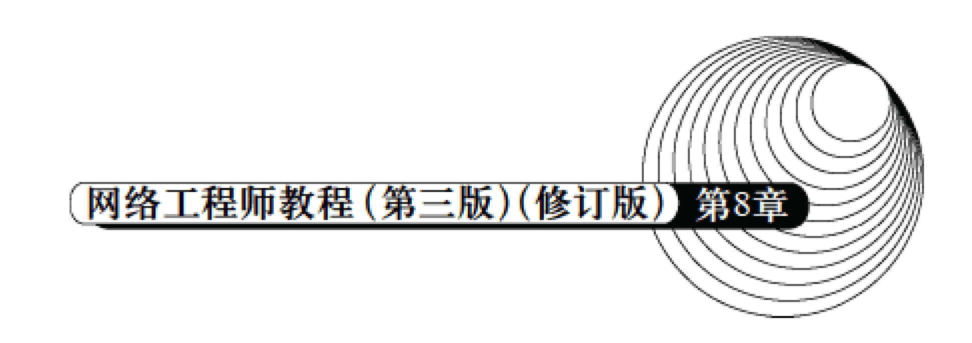
```
R2#config t
Enter configuration command, one per line. End with CNTL/Z.
R2(config)#interface E0
R2(config-if)#ip address 192.1.2.1 255.255.255.0
R2(config-if)#no keepalive
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface s0
R2(config-if)#ip address 192.168.1.2 255.255.255.0
R2(config-if)#encap frame-relay                 （该接口使用帧中继封装格式）
R2(config-if)#no shutdown
R2(config-if)#no frame-relay inverse-arp        （关闭帧中继逆向 ARP）
R2(config-if)#frame map ip 192.168.1.1 cisco
R2(config-if)#end
R1#
```

配置中关闭帧中继逆向 ARP 是因为使用了全网状拓扑，关闭逆向 ARP 是为了避免多个 DLCI 之间的映射混乱。如果在 S0 上只有一个 DLCI，则不需要该设置。

配置完成后可以用下面的命令查看帧中继相关信息，查看结果不再列出。

```
show frame pvc
show frame map
show frame traffic
show frame lmi
```

（2）配置静态路由并测试连通性。配置静态路由的方法在前面已经学习过了，配置完两个



（1）用 access-list 命令配置加密用访问控制列表。

例如：

```
access-list acl-name {permit|deny} protocol src_addr src_mask [operator port [port]] dest_addr dest_mask
[operator prot [port]]
```

（2）用 crypto ipsec transform-set 命令配置变换集。

例如：

```
crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]
```

（3）（任选）用 crypto ipsec security-association lifetime 命令配置全局性的 IPSec 安全关联的生存期。

（4）用 crypto map 命令配置加密图。

（5）用 interface 命令和 crypto map map-name interface 把配置应用到接口上。

（6）用各种可用的 show 命令验证 IPSec 的配置。

4．测试和验证 IPSec

该任务涉及到使用 show、debug 和相关的命令来测试和验证 IPSec 加密工作是否正常，并为之排除故障。

8.6.2 Cisco 配置举例

某公司由总部和分支机构构成，通过 IPSec 实现网络安全，具体网络拓扑结构和主路由器及分支路由器上的配置如下。

1．网络拓扑

网络结构如图 8-31 所示。

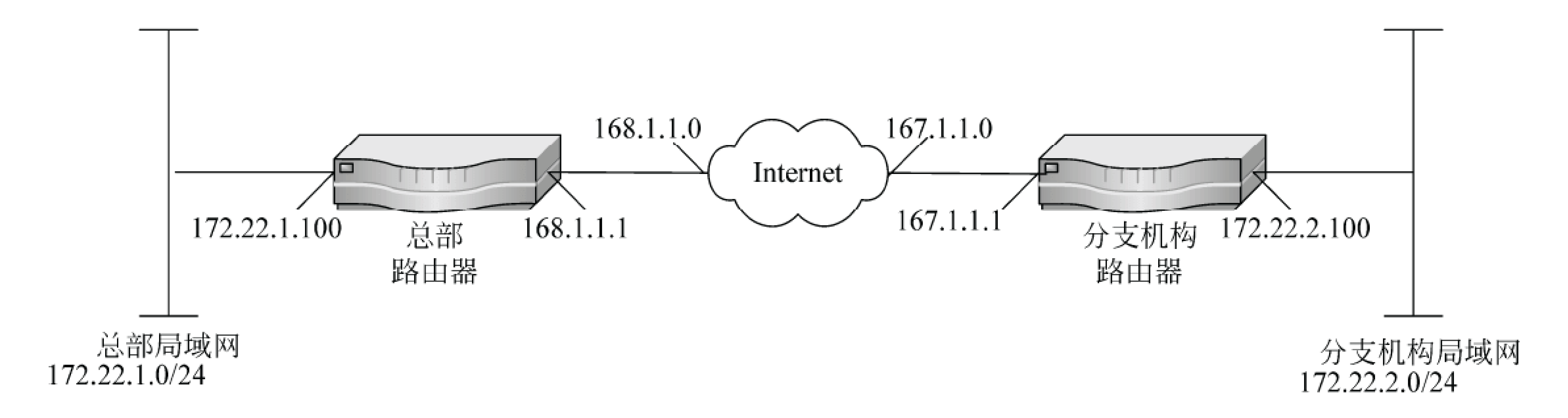


图 8-31 网络结构图

```
no ip directed-broadcast
crypto map VPNdemo ；应用 VPNdemo 于串口

!
interface Ethernet0/1
ip address 168.1.1.1 255.255.255.0        ；外部端口 IP 地址
no ip directed-broadcast
interface Ethernet0/0
ip address 172.22.1.100 255.255.255.0   ；内部端口 IP 地址
no ip directed-broadcast
!
ip classless
ip route 0.0.0.0 0.0.0.0 202.96.1.2              ；默认路由
ip route 172.22.2.0 255.255.0.0 192.168.1.2     ；到内网的静态路由（经过隧道）
access-list 101 permit gre host 202.96.1.1 host 202.96.1.2 ；定义存取列表
```

分支机构端路由器部分配置：

```
crypto isakmp policy 1
authentication pre-share
group 2
crypto isakmp key test123 address 202.96.1.1
crypto ipsec transform-set VPNtag ah-md5-hmac esp-des

!
crypto map VPNdemo 10 ipsec-isakmp
set peer 202.96.1.1
set transform-set VPNtag
match address 101

!
interface Tunnel0
ip address 192.168.1.2 255.255.255.0
no ip directed-broadcast
tunnel source Serial0/0
tunnel destination 202.96.1.1
crypto map VPNdemo

interface Serial0/0
```

```
ip address 202.96.1.2 255.255.255.252
no ip directed-broadcast
crypto map VPNdemo

!
interface Ethernet0/1
ip address 167.1.1.1 255.255.255.0
no ip directed-broadcast
interface Ethernet0/0
ip address 172.22.2.100 255.255.255.0
no ip directed-broadcast

!
ip classless
ip route 0.0.0.0 0.0.0.0 202.96.1.1
ip route 172.22.1.0 255.255.0.0 192.168.1.1
access-list 101 permit gre host 202.96.1.2 host 202.96.1.1
```

8.6.3 测试时常见的故障

1．故障 1

问题描述：在 IPSec-manual 或 IPSec-isakmp 方式下，双方配置好后或双方协商通过后，双方仍然无法进行通信。同时若打开 debug crypto packet，则会出现如下信息：

```
rec'd IPSEC packet from IPADDR has invalid spi
```

原因：对端 outbound 的 spi 值与本端的 inbound 不同或配置的策略不同（esp、ah）。

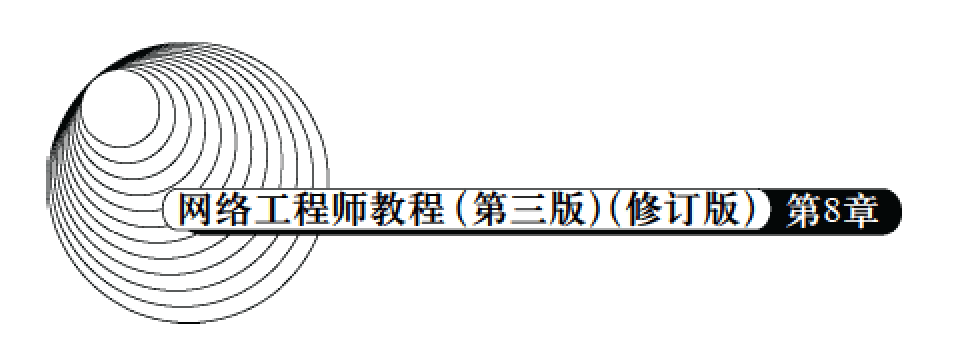
判断方法和解决方案：检查双方的配置信息，尤其是在 IPSec-manual 方式下检查双方的 SPI 值是否按方向（inbound、outbound）匹配。而在 IPSec-isakmp 下，则可能是协商出错。

2．故障 2

问题描述：在 IPSec-manual 方式下，双方配置好后，双方仍然无法进行通信。同时若打开 debug crypto packet，则会出现如下信息：

```
packet missing policy
```

原因：对端 outbound 的配置策略和本地不同（esp、ah）。



crypto isakmp，则会出现如下信息：

ISAKMP(xxx): processing ISAKMP-SA payload （随后有若干 transform-payload 中的内容）
ISAKMP(xxx): no acceptable Oakley Transform
ISAKMP(xxx)：negotiate error NO_PROPOSAL_CHOSEN

原因：双方配置的 ISAKMP 策略不匹配。

判断方法和解决方案：检查两端的 ISAKMP-Policy 是否相同，尤其是对端的 lifetime 不能大于本地的 lifetime 值。

7．故障 7

问题描述：在 IPSec-isakmp 方式下，双方配置好后，由对端开始发起协商，无法进行通信，show crypto ipsec sa 也没有发现和当前通信相关的成功的 SA 信息。在协商同时若打开 debug crypto isakmp，则会出现如下信息：

ISAKMP(xxx): processing IPSec-SA payload （随后有若干 transform-payload 中的内容）
ISAKMP(xxx): no acceptable Proposal in IPsec SA
ISAKMP(xxx)：negotiate error NO_PROPOSAL_CHOSEN

原因：双方配置的 IPSec 策略不匹配。

判断方法和解决方案：检查两端相应的 transform-set 是否匹配，相应的 sub_map 下的 pfs 属性是否相同。

8．故障 8

问题描述：在 IPSec-isakmp 方式下，双方配置好后，由对端开始发起协商，无法进行通信，show crypto ipsec sa 也没有发现和当前通信相关的成功的 SA 信息。在协商同时若打开 debug crypto isakmp，则会出现如下信息：

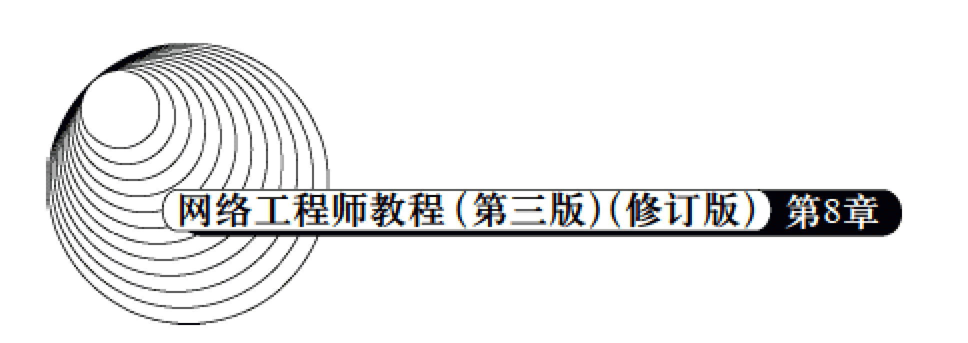
ISAKMP: attr accept again transform-set xxx ...
//ISAKMP(xxx)：dealing with ID-payload （随后有对端为 IPSec 通信所配置的 access-list 内容）
//ISAKMP(xxx): ISAKMP: not found matchable policy

原因：双方配置的 IPSec 规则不匹配。

判断方法和解决方案：检查两端相应的 sub_map 下的规则（access-list）是否匹配。

9．故障 9

问题描述：在 IPSec-isakmp 方式下，双方配置好后，由本端开始发起协商，无法进行通信，



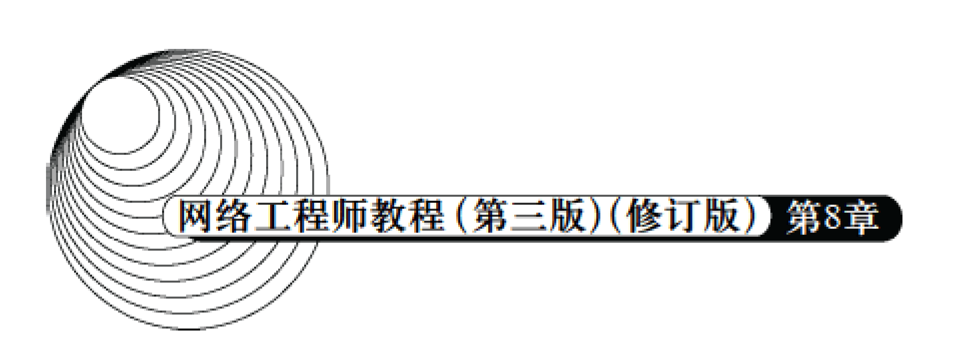
理、收发 IPv4 的分组，也可以接收、处理、收发 IPv6 的分组。对于主机来讲，“双栈”是指其可以根据需要来对业务产生的数据进行 IPv4 封装或者 IPv6 封装；对于路由器来讲，“双栈”是指在一个路由器设备中维护 IPv6 和 IPv4 两套路由协议栈，使得路由器既能与 IPv4 主机也能与 IPv6 主机通信，分别支持独立的 IPv6 和 IPv4 路由协议，IPv4 和 IPv6 路由信息按照各自的路由协议进行计算，维护不同的路由表。IPv6 数据报按照 IPv6 路由协议得到的路由表转发，IPv4 数据报按照 IPv4 路由协议得到的路由表转发。双栈策略的优点是概念清晰，易于理解，网络规划相对简单，同时在 IPv6 逻辑网络中可以充分发挥 IPv6 协议的所有优点（如安全性、路由约束和流的支持等方面）。但是，双栈策略也存在如下缺点：对网元设备的要求较高，要求其不但支持 IPv4 路由协议，而且支持 IPv6 路由协议，这就要求其维护大量的协议和数据。另外，网络升级改造将牵涉到网络中的所有网元设备，投资大、建设周期比较长。

隧道策略是 IPv4 / IPv6 过渡中经常使用到的一种机制。所谓“隧道”，简单地讲就是利用一种协议来传输另一种协议的数据的技术。在 IPv6 发展初期，必然有许多局部的纯 IPv6 网络，这些 IPv6 网络被 IPv4 骨干网络隔离开来，为了使这些孤立的“IPv6 岛”互通，就采取隧道技术的方式来解决。利用穿越现存 IPv4 因特网的隧道技术将许多个“IPv6 孤岛”连接起来，逐步扩大 IPv6 的实现范围。隧道技术的工作机理就在 IPv6 网络与 IPv4 网络间的隧道入口处，路由器将 IPv6 的数据分组封装入 IPv4 中，IPv4 分组的源地址和目的地址分别是隧道入口和出口的 IPv4 地址。在隧道的出口处再将 IPv6 分组取出转发给目的节点。目前应用较多的隧道技术包括构造隧道、6to4 隧道以及 MPLS 隧道等。目前的隧道技术主要实现了在 IPv4 数据包中封装 IPv6 数据包，随着 IPv6 技术的发展和广泛应用，未来也将会出现在 IPv4 数据包中封装 IPv6 数据包的隧道技术。隧道技术能够充分利用现有的网络投资，因此在过渡初期是一种方便的选择。但是，在隧道的入口处会出现负载协议数据包的拆分，在隧道出口处会出现负载协议数据包的重组。这就增加了隧道出入口的实现复杂度，不利于大规模的应用。

双栈策略解决了 IPv6 与 IPv4 的共存问题，但是在网络的过渡时期不可能要求所有的主机或终端都升级支持双栈，在网络中必然存在纯 IPv4 主机和纯 IPv6 主机之间进行通信的需求，由于协议栈的不同，因此很自然地需要对这些协议进行翻译转换。对应协议的翻译可以分为两个层面来进行，一方面是 IPv4 与 IPv6 协议层的翻译，另一方面是 IPv4 应用与 IPv6 应用之间的翻译。前者主要是通过 NAT－PT 技术实现的，后者则主要通过应用代理网关 ALG 来实现。NAT－PT 实现了网络层的协议翻译；应用代理网关则实现应用层的协议翻译，对于不同的应用，需要配置不同的应用代理网关。翻译技术的优点是不需要进行 IPv4、IPv6 节点的升级改造，缺点是 IPv4 节点访问 IPv6 节点的实现方法比较复杂，网络设备进行协议转换、地址转换的处理开销较大。因此，该策略一般是在其他互通方式无法使用的情况下使用。

8.7.1 IPv6-over-IPv4 GRE 隧道配置

IPv6-over-IPv4 隧道是将 IPv6 报文封装在 IPv4 报文中，让 IPv6 数据包穿过 IPv4 网络进行



1．路由器 R1 部分配置

基本配置：

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1                          （路由器命名 R1）
R1(config)#no ip domain-lookup                      （关闭路由器域名解析）
```

配置串口：

```
R1(config)# interface Serial 1/0
R1(config-if)# ip address 202.100.2.1   255.255.255.0   （设置串口地址）
R1(config-if)#no shutdown                           （开启串口）
```

配置以太口：

```
R1(config)#interface FastEthernet0/0
R1(config-if)#ipv6 address 2005:CCCC::1/64           （设置以太口地址）
R1(config-if)#exit
R1(config)#ipv6 unicast-routing                      （开启 IPv6 单播路由）
```

配置隧道：

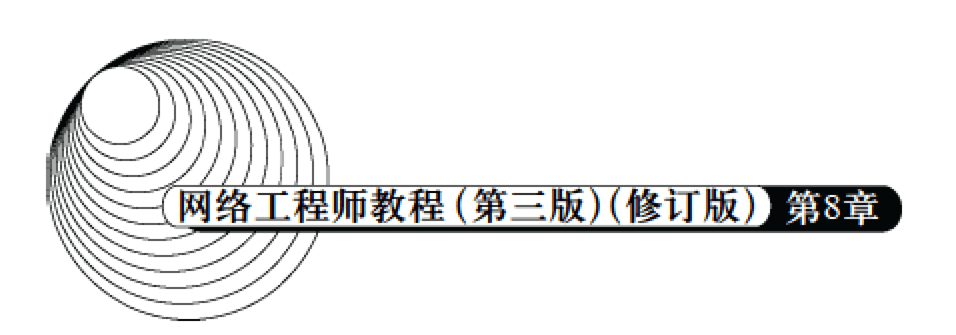
```
R1(config)#interface tunnel 0                        （启用 tunnel 0）
R1(config-if)#tunnel source s1/0                     （指定 tunnel 的源地址为 S0）
R1(config-if)#tunnel destinaltion 202.100.2.2        （指定 tunnel 的目标地址）
R1(config-if)#ipv6 address 2005:AAAA::1/64           （为 tunnel 配置 IPv6 地址）
R1(config-if)#tunnel mode gre ipv6                   （tunnel 模式为 IPv6 的 GRE 隧道）
```

配置 RIP 协议：

```
R1(config)#interface tunnel 0
R1(config-if)#ipv6 rip test enable          （在 R1 tunnel 0 上启用 RIP 协议，别名为 test）
R1(config-if)#interface FastEthernet0/0
R1(config-if)#ipv6 rip test enable          （在 R1 以太口 0 上启用 RIP 协议，别名为 test）
```

2．路由器 R2 部分配置

基本配置：



的地址——ISATAP 地址。ISATAP 地址格式为 Prefix（64bit）:0:5EFE:IPv4ADDR，其中 0: 5EFE 是 IANA 规定的格式，IPv4ADDR 是单播 IPv4 地址，它嵌入到 IPv6 地址的低 32 位。ISATAP 地址的前 64 位是通过向 ISATAP 路由器发送请求得到的，如果需要和其他网络的 ISATAP 客户端或者 IPv6 网络通信，必须通过 ISATAP 路由器拿到全球单播地址前缀(2001:, 2002:, 3ffe:开头)，通过路由器与其他 IPv6 主机和网络通信。其原理如图 8-33 所示。

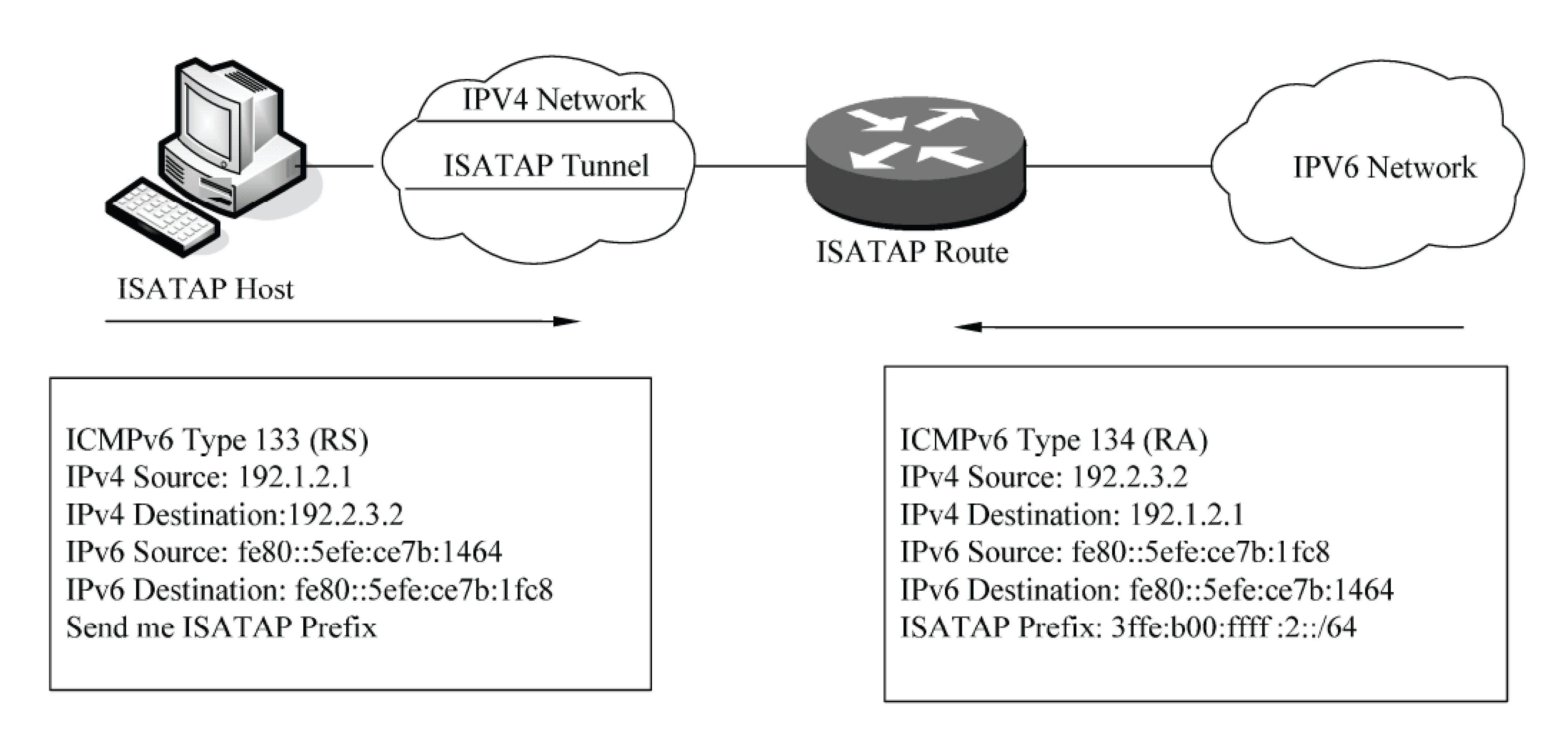


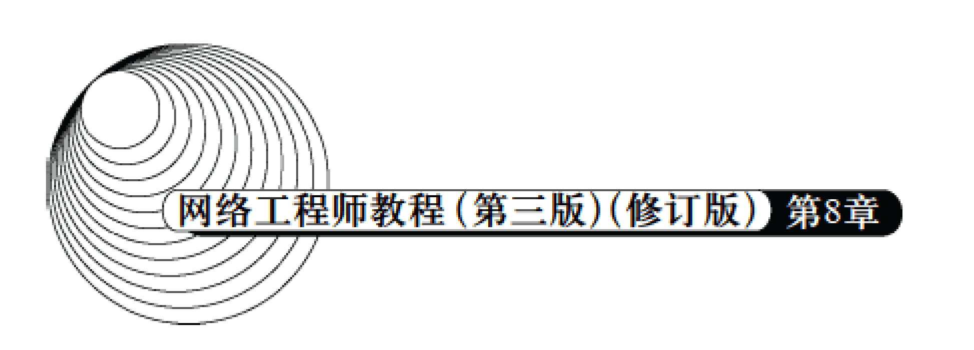
图 8-33　ISATAP 隧道获取 ISATAP 地址

ISATAP 隧道可以用于在 IPv4 网络中 IPv6 路由器—IPv6 路由器、主机—路由器的连接。由于不要求隧道节点具有全球唯一的 IPv4 地址，可以用于内部私有网络中各双栈主机进行 IPv6 通信，所以 ISATAP 隧道适用于在 IPv4 网络中 IPv6 主机之间的通信或 IPv4 网络中 IPv6 主机接入到 IPv6 网络的通信。

ISATAP 隧道相关配置命令及功能如表 8-11 所示。

表 8-11　ISATAP 隧道相关配置命令及功能

命　　令	功　　能
interface *interface-type interface-number* **ip address** *ipv4-address netmask*	给一个网络接口分配一个 IPv4 地址，这个地址被用来作为通过隧道传输的 IPv6 数据包的源 IPv4 地址，也决定了 ISATAP 路由器的 IPv6 ISATAP 地址
interface *tunnel-interface-number*	定义了路由器上启用 ISATAP 机制的隧道接口编号
tunnel source *interface-typeinterface-number*	隧道源指定了一个分配 IPv4 地址的接口。接口上的 IPv4 地址定义了分配给路由器的 ISATAP 地址的低 32 位
tunnel mode ipv6 ip isatap	确定了隧道接口的类型是 ISATAP




```
R1(config)# interface Serial 1/0
R1(config-if)# ip address 192.1.1.1    255.255.255.0         （设置串口地址）
R1(config-if)#no shutdown                                    （开启串口）
```

配置以太口：

```
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 192.0.0.1 255.255.255.0             （设置以太口地址）
R1(config-if)#exit
```

配置 ospf 协议：

```
R1(config)#router ospf 1
R1(config-router)#network 192.0.0.0 0.0.0.255 area 0
R1(config-router)#network 192.1.1.0 0.0.0.255 area 0
```

2．路由器 R3 部分配置

基本配置：

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R3                                   （路由器命名 R3）
R3(config)#no ip domain-lookup                               （关闭路由器域名解析）
```

配置串口：

```
R3(config)# interface Serial 1/0
R3(config-if)# ip address 192.2.2.1    255.255.255.0         （设置串口地址）
R3(config-if)#no shutdown                                    （开启串口）
```

配置以太口：

```
R3(config)#interface FastEthernet0/0
R3(config-if)#ipv6 address 2::1/64                           （设置以太口地址）
R3(config-if)#exit
```

配置 ospf 协议：

```
R3(config)#router ospf 1
R3(config-router)#network 192.2.2.0 0.0.0.255 area 0
```

- 动态 NAT-PT。动态模式也提供一对一的映射，但是使用一个 IPv4 地址池。池中的源 IPv4 地址数量决定了并发的 IPv6 到 IPv4 转换的最大数目。在 IPv6 网络中 IPv6 单协议网络节点动态地把预定义的 NAT-PT 前缀增加到目的 IPv4 地址。这种模式需要一个 IPv4 地址池来执行动态的地址转换，动态 NAT-PT 模式和 IPv4 中的动态 NAT 类似。
- NAPT-PT（网络地址端口转换协议转换）。NAPT-PT 提供多个有 NAT-PT 前缀的 IPv6 地址和一个源 IPv4 地址间的多对一动态映射。这种转换同时在第三层（IPv4/IPv6）和上层（TCP/UDP）进行。NAPT-PT 和 IPv4 中的 PAT 转换类似。

下面通过具体的实例来实现 NAP-PT 配置。

1．静态 NAT-PT

静态 NAT-PT 的命令及功能如表 8-12 所示。

表 8-12　静态 NAT-PT 映射命令

命　令	功　能
Router(config)#interface interface-type interface-number	指定启用 NAT-PT 机制的网络接口
Router(config-if)#ipv6 nat	在接口上启用 NAT-PT 机制，这个命令基于接口启用
Router(config)#interface interface-type interface-number	指定另外一个启用 NAT-PT 的接口
Router(config-if)#ipv6 nat	在接口上启用 NAT-PT
Router(config)# ipv6 nat prefixipv6-prefix /96	详细说明在 IPv6 域内 NAT-PT 使用的 IPv6 前缀，NAT-PT 只支持/96 的网络前缀
Router(config)#ipv6 nat v6v4 source ipv6-address ipv4-address	强制将源 IPv6 地址的输出 IPv6 数据包转换成 IPv4 数据包
Router(config)#ipv6 nat v4v6 source ipv4-address ipv6-address	强制将源 IPv4 地址的输出 IPv4 数据包转换成 IPv6 数据包

图 8-35 显示了一个静态 NAT-PT 映射配置，其中使用 2001:aaaa::2 的 IPv6 单协议网络主机可以和使用 IPv4 地址 192.17.5.2 的 IPv4 单协议网络主机通信。IPv6 网络使用 NAT-PT 网络前缀是 2001:aaaa:0:0:0:1::/96，IPv4 主机静态映射到具有 NAT-PT 前缀的 IPv6 地址 2001:aaaa: 0:0:0: 1::8，IPv6 网络主机映射到具有 NAT-PT 地址 192.17.5.200。通过在路由器上应用静态配置，IPv6 单协议网络节点和 IPv4 单协议网络节点都可以彼此通信。

R1 具体配置如下：

```
R1#configure terminal
R1(config)# interface Ethernet0
R1(config-if)#ip address 192.17.5.1 255.255.255.0
```

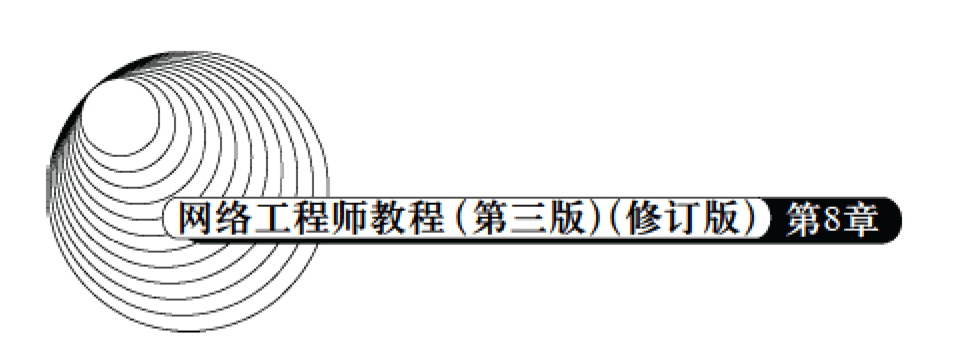


图 8-36 显示了一个动态 NAT-PT 映射配置，其中 IPv6 单协议网络 A 中的任意节点动态映射到 16.23.31.10～16.23.31.20 的地址池中的 IPv4 地址（最多 10 个主机），IPv6 单协议网络上 NAT-PT 的操作使用的前缀是 2001:b00:0:0:0:1::/96。通过在路由器 R1 上使用动态 NAT-PT 配置，IPv6 单协议网络节点可以建立到 IPv4 因特网上的 IPv4 节点的会话。

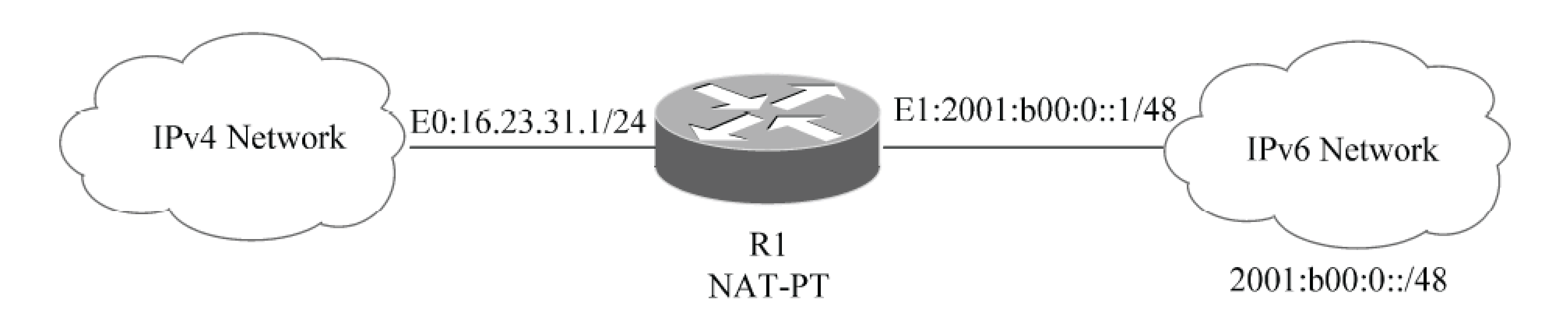


图 8-36　动态 NAT-PT 映射配置

R1 动态 NAT-PT 配置如下：

```
R1 # configure terminal
R1(config) # interface ethernet0
R1(config-if) # ip address 16.23.31.1 255.255.255.0
R1(config-if) # ipv6 nat
R1(config-if) # interface ethernet1
R1(config-if) # ipv6 address 2001:b00:0::1/48
R1(config-if) # ipv6 nat
R1(config-if) # exit
R1(config) #ipv6 access-list ipv6 permit 2001:B00:0::/48 any
R1(config) #ipv6 nat prefix 2001:b00:0:0:0:1::/96
R1(config) #ipv6 nat v6v4 pool ipv4-pool 16.23.31.10 16.23.31.20 prefix-length 24
R1(config) #ipv6 nat v6v4 source list ipv6 pool ipv4-pool
R1(config) #exit
```

3. NAPT-PT

使用 NAPT-PT 时与 PAT 转换类似，需要在配置动态 NAT-PT 映射时添加 overload 参数，参数 overload，将允许多个内部地址使用相同的全局地址，配置示例如下：

```
R1 # configure terminal
R1(config) # interface ethernet0
R1(config-if) # ip address 16.23.31.1 255.255.255.0
R1(config-if) # ipv6 nat
R1(config-if) # interface ethernet1
R1(config-if) # ipv6 address 2001:b00:0::1/48
R1(config-if) # ipv6 nat
```

当一个分组经过时，路由器按照一定的步骤找出与分组信息匹配的 ACL 语句对其进行处理。路由器自顶向下逐个处理 ACL 语句，首先把第一个语句与分组信息进行比较，如果匹配，则路由器将允许（Permit）或拒绝（Deny）分组通过；如果第一个语句不匹配，则照样处理第二个语句，直到找出一个匹配的。如果在整个列表中没有发现匹配的语句，则路由器丢弃该分组。于是，可以对 ACL 语句的处理规则总结出以下要点。

（1）一旦发现匹配的语句，就不再处理列表中的其他语句。

（2）语句的排列顺序很重要。

（3）如果整个列表中没有匹配的语句，则分组被丢弃。

需要特别强调 ACL 语句的排列顺序。如果有两条语句，一个拒绝来自某个主机的通信，另一个允许来自该主机的通信，则排在前面的语句将被执行，而排在后面的语句将被忽略。所以在安排 ACL 语句的顺序时要把最特殊的语句排在列表的最前面，而最一般的语句排在列表的最后面，这是 ACL 语句排列的基本原则。例如，下面的两条语句组成一个标准 ACL。

```
access-list 10 permit host 172.16.1.0 0.0.0.255
access-list 10 deny host 172.16.1.1
```

第一条语句表示允许来自子网 172.16.1.0/24 的所有分组通过，而第二条语句表示拒绝来自主机 172.16.1.1 的通信。如果路由器收到一个源地址为 172.16.1.1 的分组，则首先与第一条语句进行匹配，该分组被允许通过，第二条语句就被忽略了。要达到预想的结果——允许来自除主机 172.16.1.1 之外的、属于子网 172.16.1.0/24 的所有通信，则两条语句的顺序必须互换。

```
access-list 10 deny host 172.16.1.1
access-list 10 permit host 172.16.1.0 0.0.0.255
```

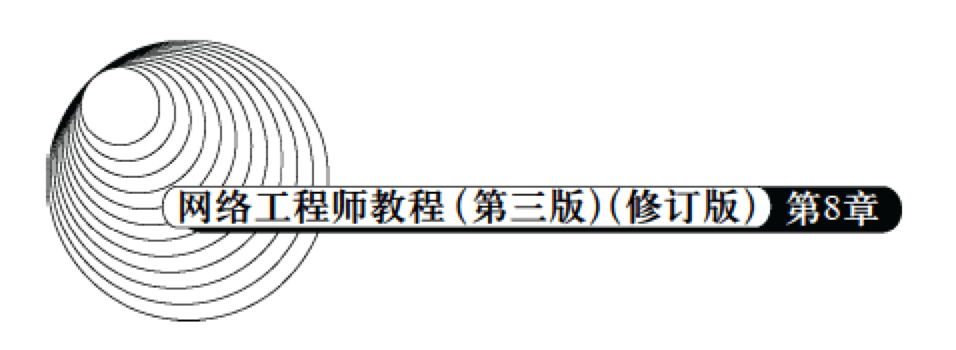
可见，列表顶上是特殊性语句，列表底部是一般性语句。

出于安全性考虑，ACL 的默认动作是拒绝（Implicit Deny），即在 ACL 中没有找到匹配的语句时分组将被拒绝通过，这相当于在列表最后有一个隐含语句拒绝了所有的通信。由此引申出的一条规则是，每一个 ACL 至少要有一条“允许”语句，否则只有“拒绝”语句的 ACL 将丢弃所有的分组。

8.8.2 ACL 配置命令

1．配置标准 ACL 的命令

Router(config)# **access-list** *ACL_#* **permit**|**deny** *conditions*



1）ACL 语句的编辑

编辑 ACL 语句有许多限制。首先，对于编号的 ACL，不能删除 ACL 中的任何表项，只能删除整个 ACL 列表。删除 ACL 的命令是：

no access-list *ACL_#*

其次，不能在 ACL 的顶部和中间插入一个表项，当前输入的 ACL 语句行总是附加在列表的底部。最后，已有的 ACL 语句也不能修改。鉴于以上限制条件，建议采用下面的过程修改一个已有的 ACL 列表。

（1）执行 show running-config 命令，找到 ACL 命令列表。

（2）复制 ACL 语句列表。

（3）粘贴在文本编辑器（例如记事本）中。

（4）对 ACL 语句进行编辑，可以插入、删除或修改 ACL 表项。

（5）复制编辑好的 ACL 文本。

（6）在路由器上执行命令 no ip access-group *ACL_#* in|out，停止 ACL 列表的应用。

（7）在路由器上执行命令 no access-list *ACL_#*，删除原来的 ACL 列表。

（8）在配置模式下粘贴编辑好的 ACL 文本。

（9）在端口配置子模式下激活新的 ACL。

2）通配符掩码

ACL 规定使用通配符掩码来说明子网地址，通配符掩码就是子网掩码按位取反的结果。如下两个特殊的通配符掩码需要说明。

- 0.0.0.0
- 255.255.255.255

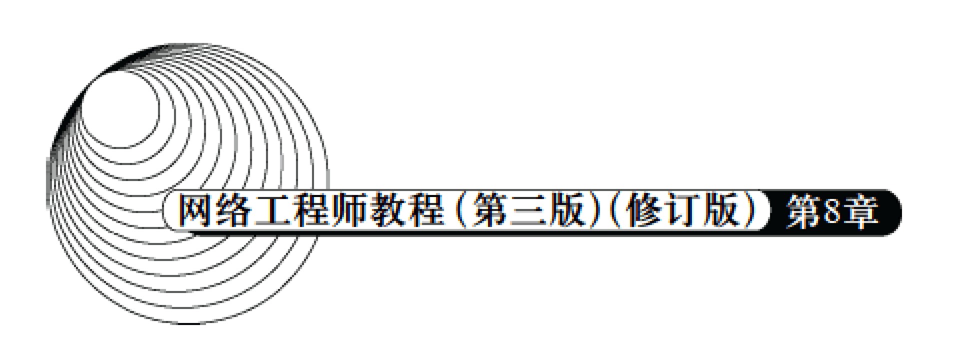
通配符掩码 0.0.0.0 表示 ACL 语句中的 32 位地址要求全部匹配，因而叫做主机掩码。例如

192.168.1.1 0.0.0.0

表示主机 192.168.1.1 的 IP 地址，实际上路由器把这个地址转换为 host 192.168.1.1，注意这里的关键字 host。通配符掩码 255.255.255.255 表示任意地址都是匹配的，通常与地址 0.0.0.0 一起使用，例如：

0.0.0.0　255.255.255.255

路由器将把这个地址转换为关键字 any。表 8-16 给出了几个使用通配符掩码的例子。




```
Router(config)# access-list 1 permit 192.168.1.0 0.0.0.255
Router(config)# interface serial 0
Router(config-if)# ip access-group 1 in
```

这样，整个 ACL 被简化为两条语句，从而提高了路由器的工作效率。

例 2 下面是配置 ACL 的另外一个例子。

```
Router(config)# access-list 2 deny 192.168.1.0
Router(config)# access-list 2 deny 172.16.0.0
Router(config)# access-list 2 permit 192.168.1.1
Router(config)# access-list 2 permit 0.0.0.0 255.255.255.255
Router(config)# interface ethernet 0
Router(config-if)# ip access-group 2 out
```

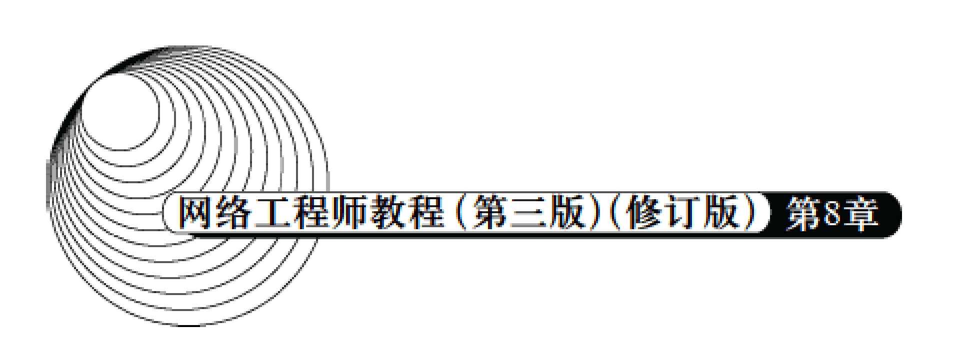
这个 ACL 存在几个问题。第一个语句似乎是拒绝了整个 192.168.1.0/24 子网发出的分组，然而实际上它什么也不做。原因是没有后随通配符掩码，这意味着要求整个地址全部匹配，然而没有一个分组的源地址会是 192.168.1.0，所以没有一个分组能够匹配这个过滤条件。第二个语句有同样的问题。第三个语句和第四个语句是对的。可见，配置 ACL 需要谨慎处理。实际上，上面的 ACL 可以修改如下：

```
Router(config)# access-list 2 deny 192.168.1.0 0.0.0.255
Router(config)# access-list 2 deny 172.16.0.0 0.0.255.255
Router(config)# access-list 2 permit 192.168.1.1
Router(config)# access-list 2 permit 0.0.0.0 255.255.255.255
Router(config)# interface ethernet 0
Router(config-if)# ip access-group 2 out
```

经过这样修改后还存在一个问题。第三个语句实际上不会执行，因为第一个语句已经拒绝了子网 192.168.1.0 发出的所有分组，所以主机 192.168.1.1 发出的分组也不会通过。要想达到预想的过滤效果，只有把特殊的语句向前提。

```
Router(config)# access-list 2 permit 192.168.1.1
Router(config)# access-list 2 deny 192.168.1.0 0.0.0.255
Router(config)# access-list 2 deny 172.16.0.0 0.0.255.255
Router(config)# access-list 2 permit any
Router(config)# interface ethernet 0
Router(config-if)# ip access-group 2 out
```

可以看出，第四个语句使用了关键字“any”，这样表示更简明。



对于 ICMP 协议，配置命令的格式为：

Router(config)# **access-list 100-199|2000-2699 permit|deny icmp**
source_address source_wildcard_mask destination_address destination_wildcard_mask
[*icmp_message*] [**log**]

其中的参数 icmp message 用于说明 ICMP 的报文类型，表 8-19 列出了几种常见的 ICMP 报文类型。

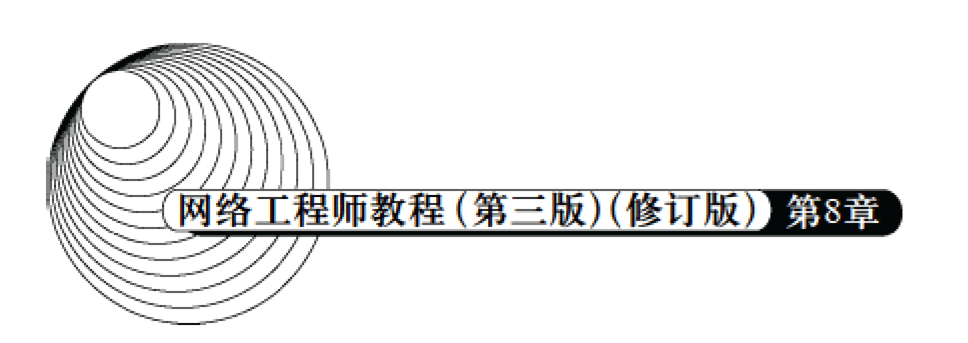
表 8-19　ICMP 报文类型

报 文 类 型	报 文 解 释
administratively-prohibited	分组被过滤，目的网络被禁止访问
echo	回声请求
echo-reply	回声响应
host-unreachable	主机不可到达
net-unreachable	网络不可到达
port-unreachable	端口不可访问
time-exceeded	分组 TTL 超时
timestamp-request	时间戳请求
timestamp-reply	时间戳响应
traceroute	Traceroute 协议（RFC1393）
mask-request	子网掩码请求
mask-reply	子网掩码响应
parameter-problem	分组参数出错
redirect	路由重定向
source-quench	源抑制

4．配置扩展 ACL 实例

例 1　下面是配置扩展 ACL 的一个例子。

Router(config)# **access-list 100 permit tcp any 172.16.0.0 0.0.255.255 established log**
Router(config)# **access-list 100 permit udp any host 172.16.1.1 eq dns log**
Router(config)# **access-list 100 permit tcp 172.17.0.0 0.0.255.255 host 172.16.1.2 eq telnet log**
Router(config)# **access-list 100 permit icmp any 172.16.0.0 0.0.255.255 echo-reply log**
Router(config)# **access-list 100 deny ip any any log**
Router(config)# **interface ethernet 0**
Router(config-if)# **ip access-group 100 in**



网络地址和路由协议的配置：

```
R1(config)# router rip
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.2.0

R2(config)# router rip
R2(config-router)# network 192.168.1.0
R1(config-router)# network 10.0.0.0

R3(config)# router rip
R3(config-router)# network 192.168.2.0
R3(config-router)# network 10.0.0.0
```

在 R1 上配置访问控制列表：

```
R1# config t
R1(config)# access-list 50 deny 10.0.0.0 0.0.0.255
R1(config)# access-list 50 permit any
R1(config)# interface fastethernet 0/0
R1(config-if)# ip access-group 50 out
R1(config-if)# ^Z
```

将 R2 和 R3 配置为 Web 服务器：

```
R2(config)# ip http server
R3(config)# ip http server
```

在 PC（192.168.3.2）上用 Web 浏览器查看 10.0.0.1 和 10.0.0.2 上的 Web 浏览器，Web 登录需要输入路由器的 enable secret 口令。

在 R1 上配置并测试访问控制列表：

```
R1(config)# access-list 101 deny tcp 192.168.3.0 0.0.0.255 10.0.0.0 0.0.0.255 eg www
R1(config)# access-list 101 deny tcp 192.168.3.0 0.0.0.255 any eg ftp
R1(config)# access-list 101 permit ip any any
R1(config)# interface fastethernet 0/0
R1(config-if)# ip access-group 101 in
R1(config-if)# ^Z
```

在 PC（192.168.3.2）上用 Web 浏览器查看 10.0.0.1 和 10.0.0.2 上的 Web 浏览器，这时无法登录。

（2）为存储管理信息提供数据库支持，例如关系数据库或面向对象的数据库。

（3）提供用户接口和用户视图（View）功能，例如管理信息浏览器。

（4）提供基本的管理操作，例如获取管理信息，配置设备参数等操作过程。

网络管理应用是用户根据需要开发的软件，这种软件运行在具体的网络上，实现特定的管理目标，例如故障诊断和性能优化，或者业务管理和安全控制等。

图 9-1 把被管理资源画在单独的框中，表明被管理资源可能与管理站处于不同的系统中。网络管理涉及到监视和控制网络中的各种硬件、固件和软件元素，例如网卡、集线器、中继器、主机、外围设备、通信软件、应用软件和实现网络互连中间件等。有关资源的管理信息由代理进程控制，代理进程通过网络管理协议与管理站对话。

9.1.2　网络管理系统的配置

网络管理系统的配置如图 9-2 所示。每一个网络节点都包含一组与管理有关的软件，叫做网络管理实体（Network Management Entity，NME）。网络管理实体完成下面的任务。

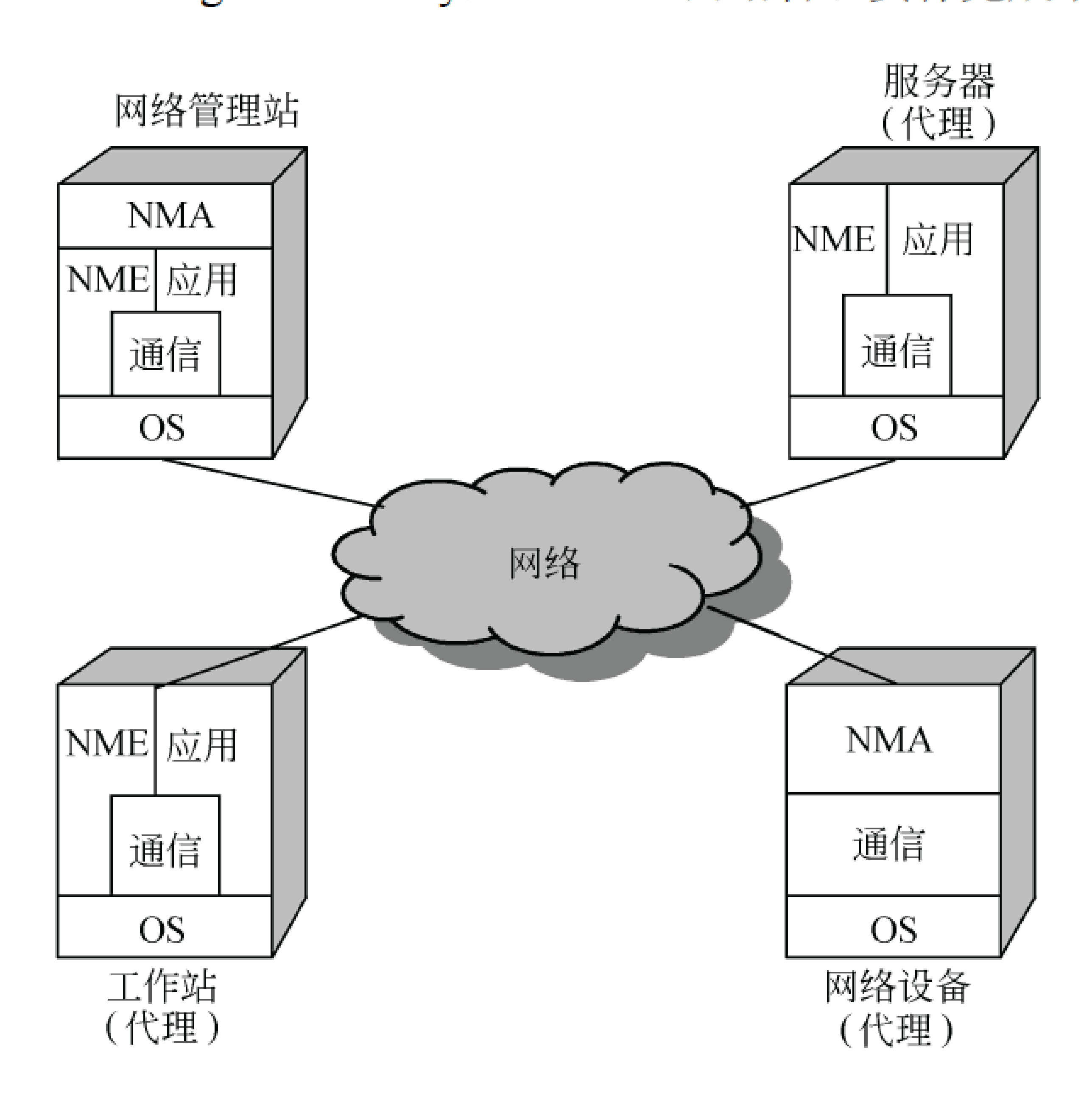
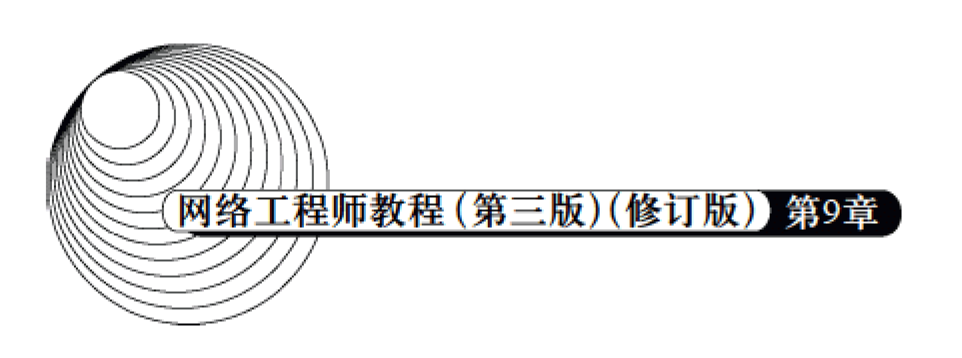


图 9-2　网络管理系统配置

（1）收集有关网络通信的统计信息。

（2）对本地设备进行测试，记录设备状态信息。

（3）在本地存储有关信息。



9.2　网络监控系统的组成

网络管理功能可分为网络监视和网络控制两大部分，统称网络监控（Network Monitoring）。网络监视是指收集系统和子网的状态信息，分析被管理设备的行为，以便发现网络运行中存在的问题。网络控制是指修改设备参数或重新配置网络资源，以便改善网络的运行状态。具体地说，网络监控要解决的问题如下。

（1）管理信息的定义。监视哪些管理信息，从哪些被管理资源获得管理信息。

（2）监控机制的设计。如何从被管理资源得到需要的信息。

（3）管理信息的应用。根据收集到的管理信息实现什么管理功能。

下面首先说明前两个问题，即管理信息的定义和监控机制。

9.2.1　管理信息的组成

对网络监控有用的管理信息可以分为如下三类。

- 静态信息。包括系统和网络的配置信息，例如路由器的端口数和端口编号，工作站的标识和 CPU 类型等，这些信息不经常变化。
- 动态信息。与网络中出现的事件和设备的工作状态有关，例如网络中传送的分组数、网络连接的状态等。
- 统计信息。即从动态信息推导出的信息，例如平均每分钟发送的分组数、传输失败的概率等。

这些信息组成的管理信息库如图 9-6 所示。配置数据库中存储着计算机和网络的基本配置信息，传感器数据库中存储着传感器的设置信息。传感器是一组软件，用于实时地读取被管理设备的有关参数。配置数据库和传感器数据库共同组成静态数据库。动态数据库存储着由传感器收集的各种网络元素和网络事件的实时数据。统计数据库中的管理信息是由动态信息计算出来的。图 9-6 表示出这三种数据库的关系。

网络监控功能一方面要确定从哪里收集管理信息，另一方面还要确定管理信息应该存储在什么地方。静态信息是由网络元素直接产生的，通常由驻留在这些网络元素（例如路由器）中的代理进程收集和存储，必要时传送给监视器。如果网络元素（例如 Modem）中没有代理进程，则可以由委托代理收集这些静态信息，并传送给监视器。

动态信息通常也是由产生有关事件的网络元素收集和存储的。例如，工作站建立的网络连接数就存储在该工作站中。然而对于一个局域网来说，网络中各个设备的行为和有关数据可以由连接在网络中的一个专用主机来收集和记录，这个主机叫做远程网络监视器，它的作用是收集整个子网的通信数据，例如在一段时间内一对主机交换的分组数，或网络中出现的冲突次

数等。

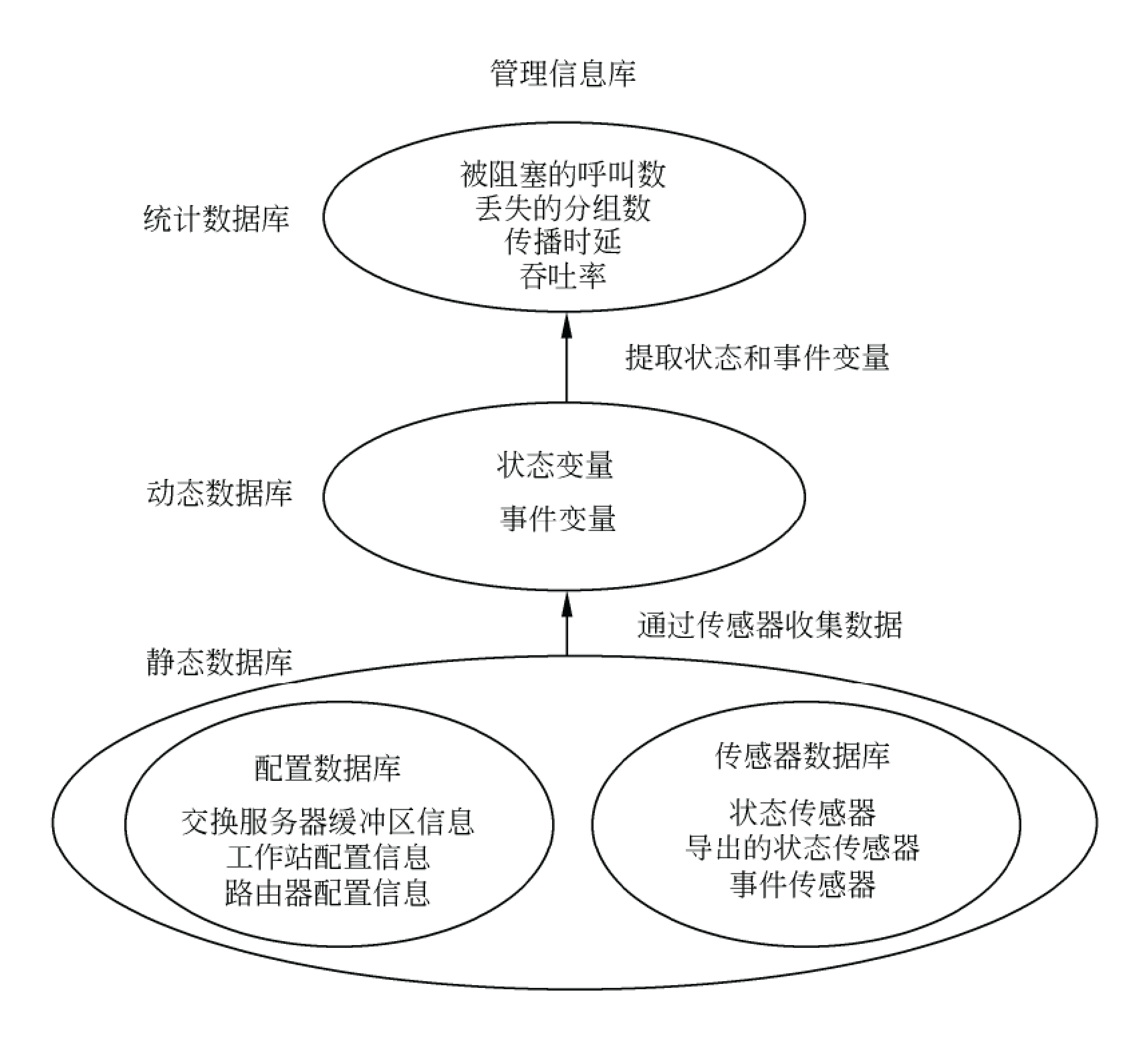


图 9-6　管理信息库的组成

统计信息可以由任何能够访问动态信息的系统产生。当然，统计信息也可以由网络监视器自己产生，这就要求把所有需要的原始数据传送给监视器，再由监视器进行分析和计算。如果原始数据的量很大，则这种监控方式可能会消耗很多网络带宽。如果存储动态信息的系统进行了分析和计算，则不但节约了网络带宽，而且也节省了监视器的处理时间。

9.2.2　网络监控系统的配置

网络监控系统的配置如图 9-7（a）所示。监控应用程序是监控系统的用户接口，它完成性能监视、故障监视和计费监视等功能。管理功能负责与其他网络元素中的代理进程通信，把需要的监控信息提供给监控应用程序。这两个模块都处于管理站中。管理对象表示被监控的网络资源中的管理信息，所有管理对象遵从网络管理标准的规定。管理对象中的信息通过代理功能提供给管理站。图 9-7（b）中增加了监控代理功能。这个模块的作用是专门对管理信息进行计算和统计分析，并且把计算的结果提供给管理站。在管理站看来，监控代理的作用和一般代理

报告。

事件报告是由代理主动发送给管理站的消息。代理可以根据管理站的要求（周期，内容等）定时地发送状态报告，也可能在检测到某些特定事件（例如状态改变）或非正常事件（例如出现故障）时生成事件报告，发送给管理站。事件报告对于及时发现网络中的问题是很有用的，特别是对于监控状态信息不经常改变的管理对象更有效。

在已有的各种网络监控系统中都设置了轮询和事件报告两种通信机制，但强调的重点有所不同。传统的通信管理网络主要依赖事件报告，而 SNMP 强调轮询方法，OSI 系统管理则采取了这两种极端方法的中间道路。然而无论是 SNMP，或是 OSI，以及某些专用的管理系统都允许用户根据具体情况决定使用何种通信方式。影响通信方式选择的主要因素如下。

（1）传送监控信息需要的通信量。

（2）对危急情况的处理能力。

（3）对网络管理站的通信时延。

（4）被管理设备的处理工作量。

（5）消息传输的可靠性。

（6）网络管理应用的特殊性。

（7）在发送消息之前通信设备失效的可能性。

9.3 网络管理功能域

网络管理有 5 大功能域，即故障管理（Fault Management）、配置管理（Configuration Management）、计费管理（Accounting Management）、性能管理（Performance Management）和安全管理（Security Management），简写为 F-CAPS。传统上，性能、故障和计费管理属于网络监视功能，另外两种属于网络控制功能。

9.3.1 性能管理

网络监视中最重要的是性能监视，然而要能够准确地测量出对网络管理有用的性能参数却是不容易的。可选择的性能指标很多，有些很难测量，或计算量很大，但不一定很有用；有些有用的指标则没有得到制造商的支持，无法从现有的设备上检测到。还有些性能指标互相关联，要互相参照才能说明问题。这些情况都增加了性能测量的复杂性。这一小节介绍性能管理的基本概念，给出对网络管理有用的两类性能指标：面向服务的性能指标和面向效率的性能指标。当然，网络最主要的目标是向用户提供满意的服务，因而面向服务的性能指标应具有较高的优先级。下面前三个是面向服务的性能指标，后两个是面向效率的性能指标。

路工作的概率为 $A（1-A）+（1-A）A=2A-2A^2=0.18$。则有

$$A_f（非峰值时段）=1.0\times0.18+1.0\times0.81=0.99$$

$$A_f（峰值时段）=0.8\times0.18+1.0\times0.81=0.954$$

于是系统的平均可用性为

$$A_f=0.6\times A_f（峰值时段）+0.4\times A_f（非峰值时段）=0.9684$$

2. 响应时间

响应时间是指从用户输入请求到系统在终端上返回计算结果的时间间隔。从用户角度看，这个时间要和人们的思考时间（等于两次输入之间的最小间隔时间）配合，越是简单的工作（例如数据录入）要求响应时间越短。然而从实现角度看，响应时间越短，实现的代价越大。研究表明，系统响应时间对人的生产率的影响是很大的。在交互式应用中，响应时间大于 15s，大多数人是不能容忍的。响应时间大于 4s 时，人们的短期记忆会受到影响，工作的连续性会被破坏。尤其是对数据录入人员来说，这种情况下击键的速度会严重受挫，只是在输入完一个段落后，才可以有比较大的延迟（例如 4s 以上）。越是注意力高度集中的工作，要求响应时间越短。特别是对于需要记住以前的响应、根据前边的响应决定下一步的输入时，延迟时间应该小于 2s。在用鼠标单击图形或进行键盘输入时，要求的响应时间更小，可能在 0.1s 以下。这样人们会感到计算机是同步工作的，几乎没有等待时间。图 9-9 表示应用 CAD 进行集成电路设计时生产

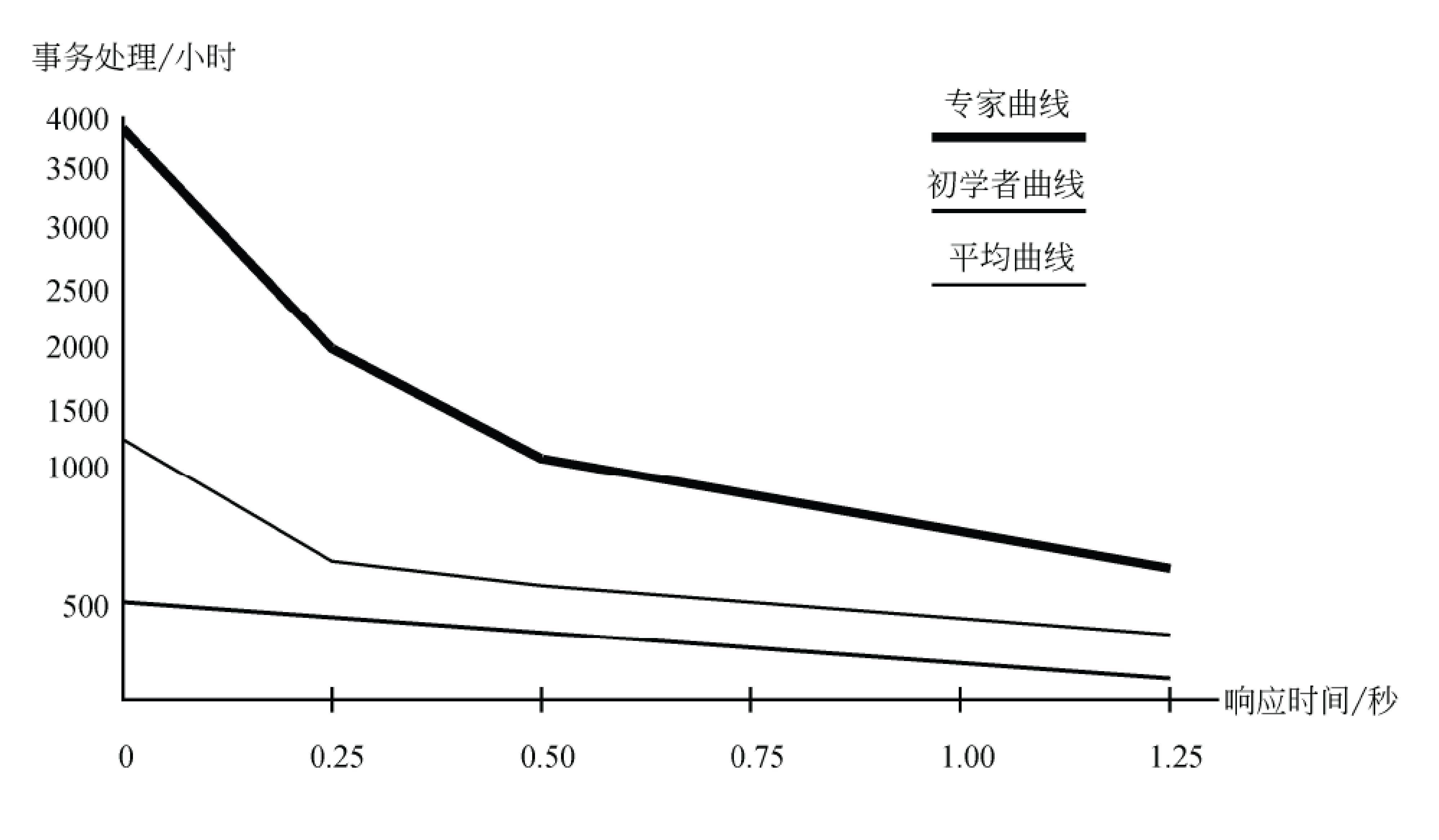


图 9-9 系统响应时间与生产率的关系

- 出口服务时间：通过网络把响应报文传送到网络接口设备的处理时间。
- 出口终端延迟：终端接收响应报文的时间，主要是由通信延迟引起的。

响应时间是比较容易测量的，是网络管理中重要的管理信息。

3. 正确性

这是指网络传输的正确性。由于网络中有内置的纠错机制，所以通常用户不必考虑数据传输是否正确。但是，监视传输误码率可以发现瞬时的线路故障，以及是否存在噪声源和通信干扰，以便及时采取维护措施。

4. 吞吐率

吞吐率是面向效率的性能指标，具体表现为一段时间内完成的数据处理量（Mbps 或分组数每秒），或者接受用户会话的数量，或者处理呼叫的数量等。跟踪这些指标可以为提高网络传输效率提供依据。

5. 利用率

利用率是指网络资源利用的百分率，它也是面向效率的指标。这个参数与网络负载有关，当负载增加时，资源利用率增大，因而分组排队时间和网络响应时间变长，甚至会引起吞吐率降低。当相对负载（负载/容量）增加到一定程度时，响应时间迅速增长，从而引发传输瓶颈和网络拥挤。图 9-11 表示响应时间随相对负载成指数上升的情况。特别值得注意的是，实际情况往往与理论计算结果相左，造成失去控制的通信阻塞，这是应该设法避免的，所以需要更精致的分析技术。

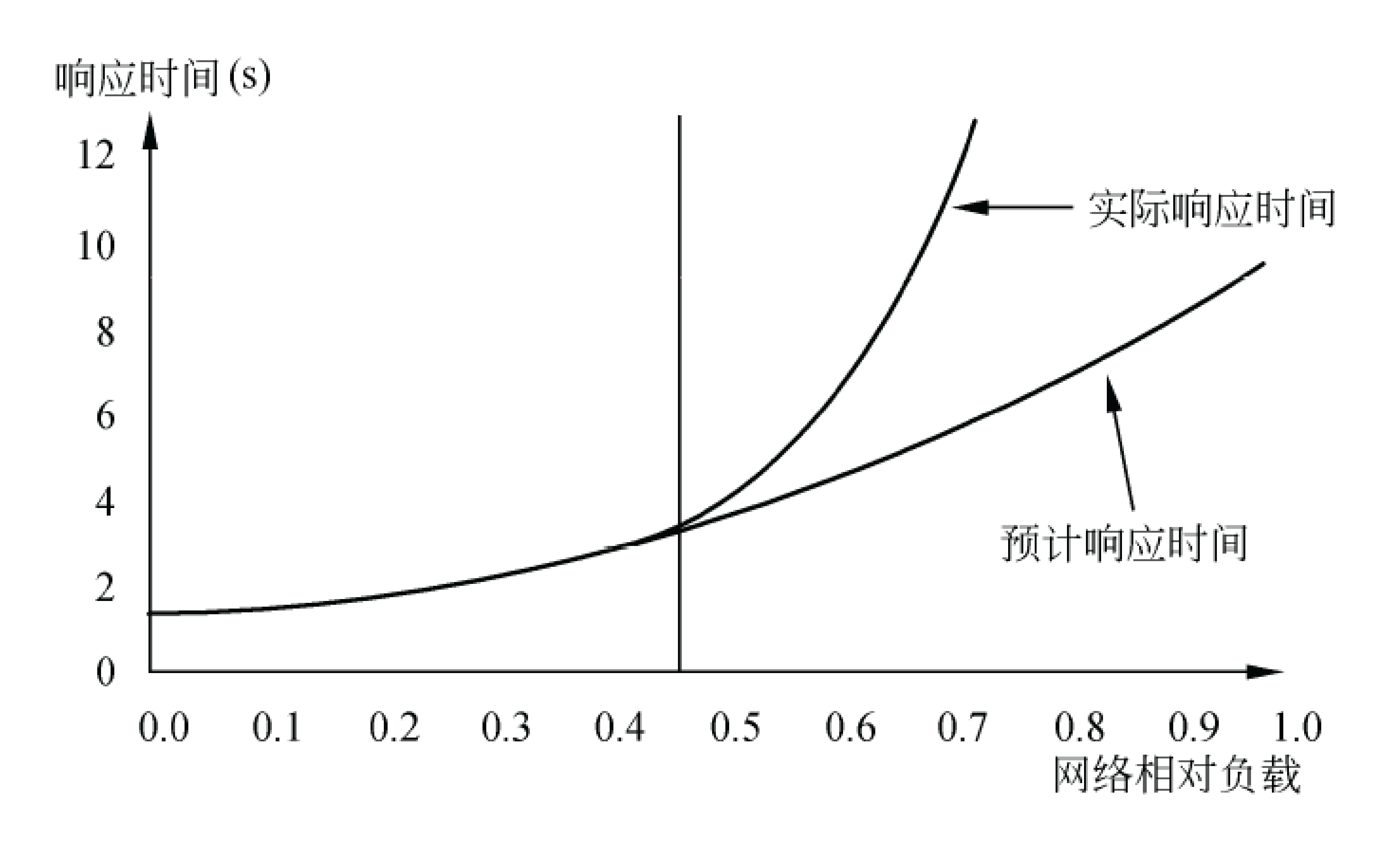


图 9-11　网络响应时间与负载的关系

收集到的性能参数组织成性能测试报告，以图形或表格的形式呈现给网络管理员。对于局域网来说，性能测试报告应包括：

- 主机对通信矩阵。一对源主机和目标主机对之间传送的总分组数、数据分组数、数据字节数以及它们所占的百分比。
- 主机组通信矩阵。一组主机之间通信量的统计，内容与上一条类似。
- 分组类型直方图。各种类型的原始分组（例如广播分组、组播分组等）的统计信息，用直方图表示。
- 数据分组长度直方图。不同长度（字节数）的数据分组的统计。
- 吞吐率——利用率分布。各个网络节点发送/接收的总字节数和数据字节数的统计。
- 分组到达时间直方图。不同时间到达的分组数的统计。
- 信道获取时间直方图。在网络接口单元（NIU）排队等待发送、经过不同延迟时间的分组数的统计。
- 通信延迟直方图。从发出原始分组到分组到达目标的延迟时间的统计。
- 冲突计数直方图。经受不同冲突次数的分组数的统计。
- 传输计数直方图。经过不同试发送次数的分组数的统计。

另外，还应包括功能全面的性能评价程序（对网络当前的运行状态进行分析）和人工负载生成程序（产生性能测试数据），帮助管理人员进行管理决策。

9.3.2 故障管理

故障监视就是要尽快地发现故障，找出故障原因，以便及时采取补救措施。在复杂的系统中，发现和诊断故障是不容易的。首先是有些故障很难观察到，例如分布处理中出现的死锁就很难发现。其次是有些故障现象不足以表明故障原因，例如发现远程节点没有响应，但是否低层通信协议失效不得而知。更有些故障现象具有不确定性和不一致性，引起故障的原因很多，使得故障定位复杂化。例如，终端死机、线路中断、网络拥塞或主机故障都会引起同样的故障现象，到底问题出在哪儿，需要复杂的故障定位手段。故障管理可分为如下三个功能模块。

（1）故障检测和报警功能。故障监视代理要随时记录系统出错的情况和可能引起故障的事件，并把这些信息存储在运行日志数据库中。在采用轮询通信的系统中，管理应用程序定期访问运行日志记录，以便发现故障。为了及时检测重要的故障问题，代理也可以主动向有关管理站发送出错事件报告。另外，对出错报告的数量、频率要有适当地控制，以免加重网络负载。

（2）故障预测功能。对各种可以引起故障的参数建立门限值，并随时监视参数值变化，一旦超过门限值，就发送警报。例如，由于出错产生的分组碎片数超过一定值时发出警报，表示线路通信恶化，出错率上升。

9.3.4 配置管理

配置管理是指初始化、维护和关闭网络设备或子系统。被管理的网络资源包括物理设备（例如服务器、路由器）和底层的逻辑对象（例如传输层定时器）。配置管理功能可以设置网络参数的初始值/默认值，使网络设备初始化时自动形成预定的互联关系。当网络运行时，配置管理监视设备的工作状态，并根据用户的配置命令或其他管理功能的请求改变网络配置参数。例如，若性能管理检测到响应时间延长，并分析出性能降级的原因是由于负载失衡，则配置管理将通过重新配置（例如改变路由表）改善系统响应时间。又例如，故障管理检测到一个故障，并确定了故障点，则配置管理可以改变配置参数，把故障点隔离，恢复网络正常工作。配置管理应包含下列功能模块。

- 定义配置信息。
- 设置和修改设备属性。
- 定义和修改网络元素间的互联关系。
- 启动和终止网络运行。
- 发行软件。
- 检查参数值和互联关系。
- 报告配置现状。

最后两项属于配置监视功能，即管理站通过轮询随时访问代理保存的配置信息，或者代理通过事件报告及时向管理站通知配置参数改变的情况。下面解释配置控制的其他功能。

1. 定义配置信息

配置信息描述网络资源的特征和属性，这些信息对其他管理功能是有用的。网络资源包括物理资源（例如主机、路由器、网桥、通信链路和 Modem 等）和逻辑资源（例如定时器、计数器和虚电路等）。设备的属性包括名称、标识符、地址、状态、操作特点和软件版本。配置信息可以有多种组织方式。简单的配置信息组织成由标量组成的表，每一个标量值表示一种属性值，SNMP 采用这种方法。在 OSI 系统管理中，管理信息定义为面向对象的数据库。对象的值表示被管理设备的特性，对象的行为（例如通知）代表了管理操作，对象之间的包含关系和继承关系则规范了它们之间的互相作用。另外，还有一些系统用关系数据库表示管理信息。

管理信息存储在与被管理设备最接近的代理或委托代理中，管理站通过轮询或事件报告访问这些信息。网络管理员可以在管理站提供的用户界面上说明管理信息值的范围和类型，用以设置被管理资源的属性。网络控制功能还允许定义新的管理对象，在指定的代理中生成需要的管理对象或数据元素。产生新数据的过程可以是联机的、动态的，或是脱机的、静态的。

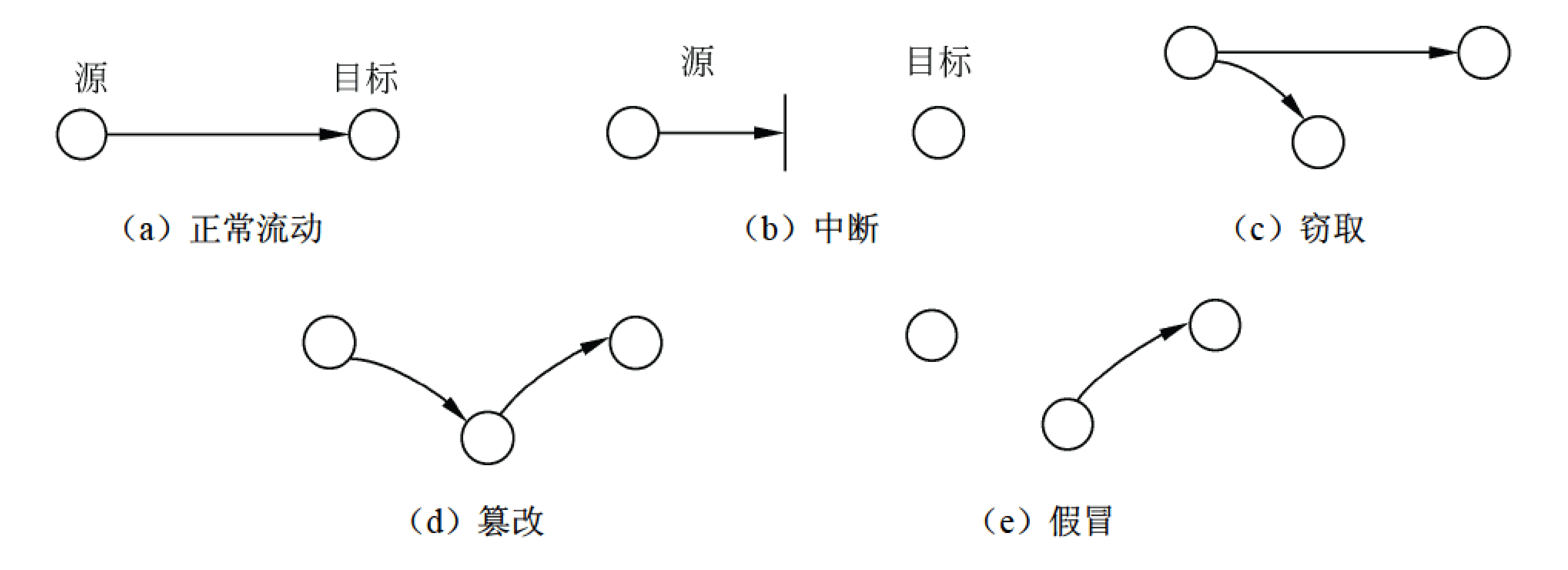


图 9-13　对网络通信的安全威胁

（d）篡改（modification）。未经授权的入侵者不仅访问了信息资源，而且篡改了信息，这是对数据完整性的威胁。例如改变文件中的数据，改变程序的功能，修改网上传送的报文等。

（e）假冒（fabrication）。未经授权的入侵者在网络信息中加入了伪造的内容，这也是对数据完整性的威胁。例如向网络用户发送虚假的消息，在文件中插入伪造的记录等。

2．对计算机网络的安全威胁

图 9-14 所示为对计算机网络的各种安全威胁，分别解释如下。

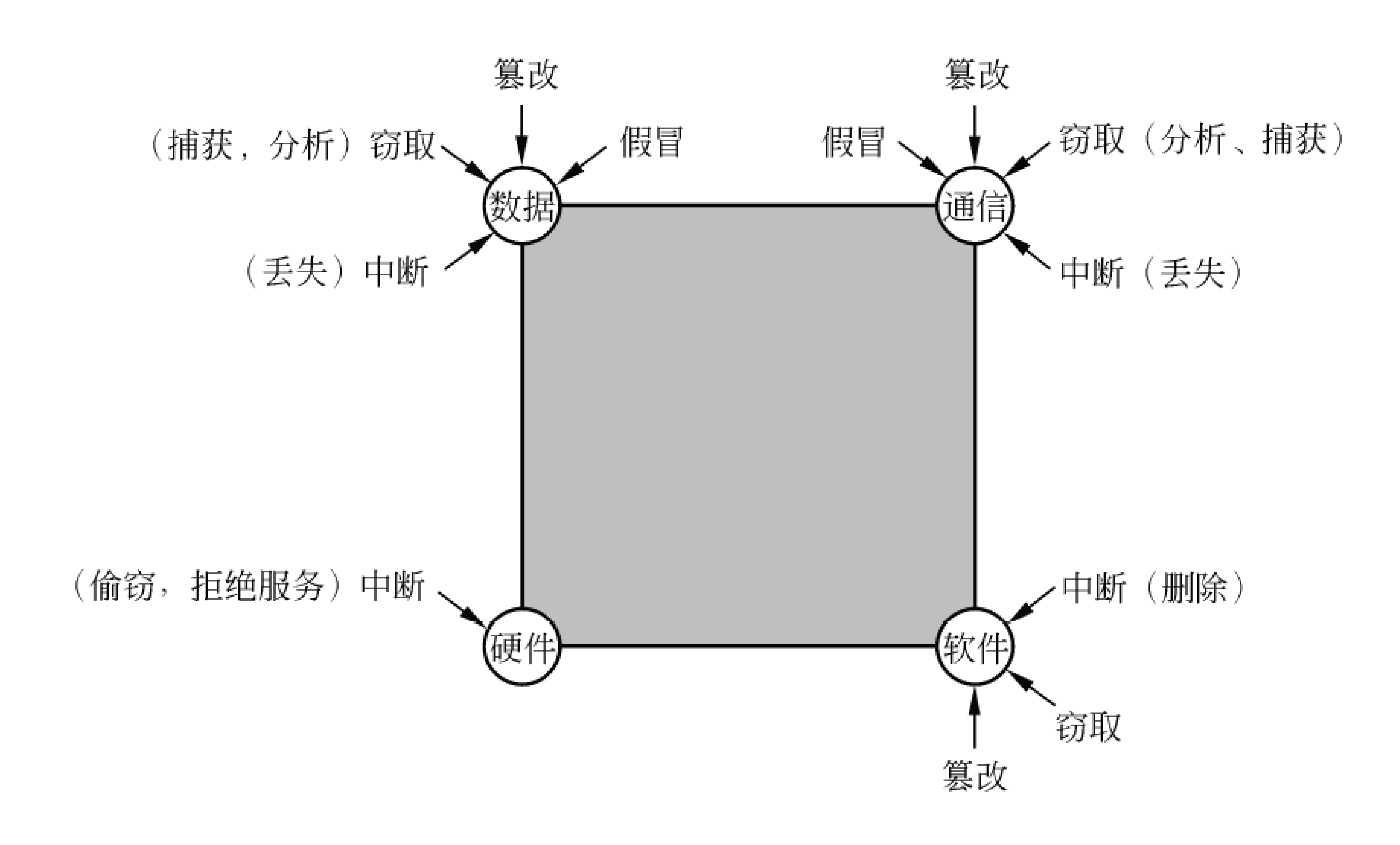


图 9-14　对计算机网络资源的安全威胁

（1）对硬件的威胁。主要是破坏系统硬件的可用性，例如有意或无意的损坏、甚至盗窃网

安全管理记录用户的活动属性（Profile）以及特殊文件的使用属性，检查可能出现的异常访问活动。安全管理功能使管理人员能够生成和删除与安全有关的对象，改变它们的属性或状态，影响它们之间的关系。

3．加密过程控制

安全管理能够在必要时对管理站和代理之间交换的报文进行加密。安全管理也能够使用其他网络实体的加密方法。此外，这个功能还可以改变加密算法，具有密钥分配能力。

9.4 简单网络管理协议

在 20 世纪 80 年代末，随着对网络管理系统的迫切需求和网络管理技术的日臻成熟，国际标准化组织开始制订关于网络管理的国际标准。首先是 ISO 在 1989 年颁布了 ISO DIS 7498-4（X.700）文件，定义了网络管理的基本概念和总体框架，后来在 1991 年发布的两个文件中规定了网络管理提供的服务和网络管理协议，即 ISO 9595 公共管理信息服务定义（Common Management Information Service，CMIS）和 ISO 9596 公共管理信息协议规范（Common Management Information Protocol，CMIP）。在 1992 年公布的 ISO 10164 文件中规定了系统管理功能（System Management Functions，SMFs），而 ISO 10165 文件则定义了管理信息结构（Structure of Management Information，SMI）。这些文件共同组成了 ISO 的网络管理标准。这是一个非常复杂的协议体系，管理信息采用了面向对象的模型，管理功能包罗万象，另外还有一些附加的功能和一致性测试方面的说明。由于其复杂性，有关 ISO 管理的实现进展缓慢，很少有适用的网管产品。

另一方面，随着 20 世纪 90 年代初 Internet 的迅猛发展，有关 TCP/IP 网络管理的研究活动十分活跃，另一类网络管理标准正在迅速流传和广泛应用。TCP/IP 网络管理方面最初使用的是 1987 年 11 月提出的简单网关监控协议（Simple Gateway Monitoring Protocol，SGMP），在此基础上改进成简单网络管理协议第一版（Simple Network Management Protocol，SNMPv1），陆续公布在 1990 和 1991 年的几个 RFC（Request For Comments）文件中，即 RFC 1155（SMI）、RFC1157（SNMP）、RFC1212（MIB 定义）和 RFC1213（MIB-2 规范）。由于其简单性和易于实现，SNMPv1 得到了许多制造商的支持和广泛的应用。几年以后，在第一版的基础上改进功能和安全性，又产生了第二版 SNMPv2（RFC1902-1908，1996）和 SNMPv3（RFC2570-2575 Apr.1999）。

在同一时期，用于监控局域网通信的标准——远程网络监控（Remote Monitoring，RMON）也出现了，这就是 RMON-1（1991）和 RMON-2（1995）。这一组标准定义了监视网络通信的管理信息库，是 SNMP 管理信息库的扩充，与 SNMP 协议配合可以提供更有效的管理性能，也得到了广泛应用。

另外，IEEE 定义了局域网的管理标准，即 IEEE 802.1b LAN/MAN 管理。这个标准用于管理物理层和数据链路层的 OSI 设备，因而叫做 CMOL（CMIP over LLC）。

为了适应电信网络的管理需要，ITU-T 在 1989 年定义了电信网络管理标准（Telecommunications Management Network，TMN），即 M.30 建议（蓝皮书）。

9.4.1　SNMPv1

Internet 最初的网络管理框架由 4 个文件定义，如图 9-16 所示，这就是 SNMPv1。RFC1155 定义了管理信息结构，规定了管理对象的语法和语义。SMI 主要说明了怎样定义管理对象和怎样访问管理对象。RFC1212 说明了定义 MIB 模块的方法，而 RFC1213 则定义了 MIB-2 管理对象的核心集合，这些管理对象是任何 SNMP 系统必须实现的。最后，RFC1157 是 SNMPv1 协议的规范文件。

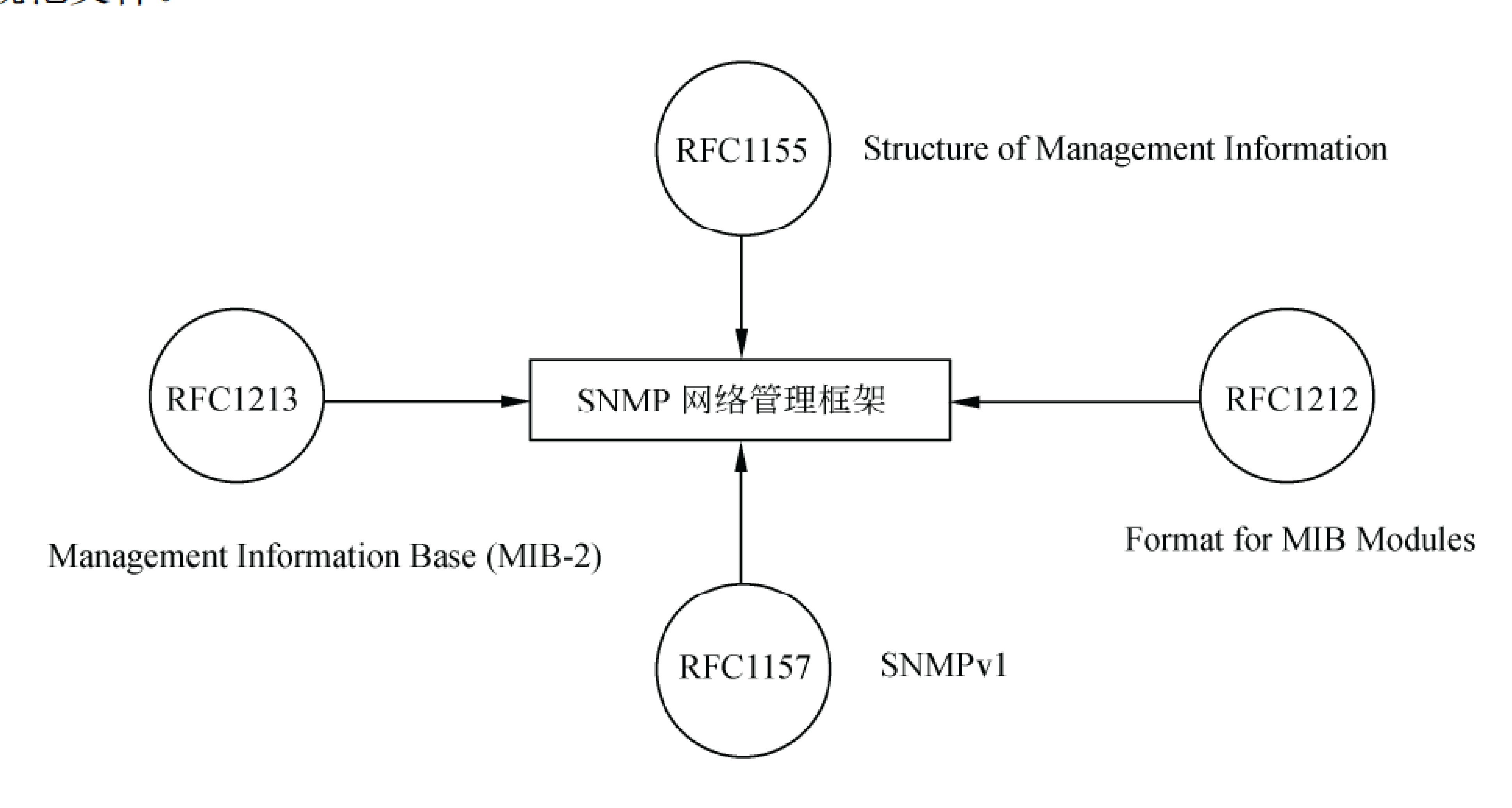
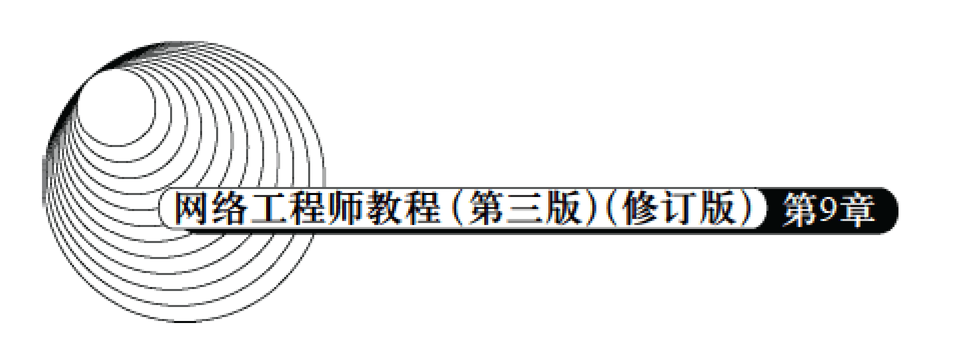


图 9-16　SNMPv1 网络管理框架的定义

1. SNMP 体系结构

图 9-17 所示为 Internet 网络管理的体系结构。由于 SNMP 定义为应用层协议，所以它依赖于 UDP 数据报服务。同时，SNMP 实体向管理应用程序提供服务，它的作用是把管理应用程序的服务调用变成对应的 SNMP 协议数据单元，并利用 UDP 数据报发送出去。

其所以选择 UDP 协议而不是 TCP 协议，是因为 UDP 效率较高，这样实现网络管理不会太多地增加网络负载。但由于 UDP 不是很可靠，所以 SNMP 报文容易丢失。为此，对 SNMP 实现的建议是对每个管理信息要装配成单独的数据报独立发送，而且报文应短些，不要超过 484



议数据单元（PDU）。报文头中的版本号是指 SNMP 的版本，RFC1157 为第一版。团体名用于身份认证。SNMP 共有 5 种管理操作，但只有 4 种 PDU 格式。管理站发出的三种请求报文 GetRequest、GetNextRequest 和 SetRequest 采用的格式是一样的，代理的应答报文格式只有一种 GetResponsePDU。关于 PDU 中各个字段的含义，解释如下。

SNMP 报文

版本号	团体名	SNMP PDU

GetRequestPDU, GetNextRequestPDU 和 SetRequestPDU

PDU 类型	请求标识	0	0	变量绑定表

GetResponsePDU

PDU 类型	请求标识	错误状态	错误索引	变量绑定表

TrapPDU

PDU 类型	制造商ID	代理地址	一般陷入	特殊陷入	时间戳	变量绑定表

变量绑定表

名字 1	值 1	名字 2	值 2	…	名字 *n*	值 *n*

图 9-19　SNMP 报文格式

从图 9-19 中可以看出，除了 Trap 之外的 4 种 PDU 格式是相同的，共有 5 个字段。

- PDU 类型：共 5 种类型的 PDU。
- 请求标识（request-id）：赋予每个请求报文唯一的整数，用于区分不同的请求。由于在具体实现中请求多是在后台执行，当应答报文返回时要根据其中的请求标识与请求报文配对。请求标识的另一个作用是检测由不可靠的传输服务产生的重复报文。
- 错误状态（error-status）：表示代理在处理管理站的请求时可能出现的各种错误。
- 错误索引（error-index）：当错误状态非 0 时指向出错的变量。
- 变量绑定表（variable-binding）：变量名和对应值的表，说明要检索或设置的所有变量及其值。在检索请求报文中，变量的值应为 0。

3. SNMP 协议的操作

SNMP 报文在管理站和代理之间传送，包含 GetRequest、GetNextRequest 和 SetRequest 的报文由管理站发出，代理以 GetResponse 响应。所有报文发送和应答序列如图 9-20 所示。一般来说，管理站可连续发出多个请求报文，然后等待代理返回的应答报文。如果在规定的时间内收到应答，则按照请求标识进行配对，即应答报文必须与请求报文有相同的请求标识。

允许 Get 和 Trap 操作，通过 Set 操作控制网络设备是被严格限制的。

SNMP 定义的陷入类型是很少的，虽然可以补充设备专用的陷入类型，但专用的陷入往往不能被其他制造商的管理站理解，所以管理站主要靠轮询收集信息。轮询的频率对管理的性能影响很大。如果管理站在启动时轮询所有代理，以后只是等待代理发来的陷入，这样就很难掌握网络的最新动态。例如，不能及时了解网络中出现的拥塞。

需要一种能提高网络管理性能的轮询策略，以决定合适的轮询频率。通常轮询频率与网络的规模和代理的多少有关。而网络管理性能还取决于管理站的处理速度、子网数据速率、网络拥塞程度等众多的因素，所以很难给出准确的判断规则。为了使问题简化，假定管理站一次只能与一个代理作用，轮询只是采用 get 请求/响应这种简单形式，而且管理站全部时间都用来轮询，于是有下面的不等式

$$N \leqslant T / \Delta$$

其中：N——被轮询的代理数；

T——轮询间隔；

Δ——单个轮询需要的时间。

Δ 与下列因素有关。

（1）管理站生成一个请求报文的时间。

（2）从管理站到代理的网络延迟。

（3）代理处理一个请求报文的时间。

（4）代理产生一个响应报文的时间。

（5）从代理到管理站的网络延迟。

（6）管理站处理一个响应报文的时间。

（7）为了得到需要的管理信息，交换请求/响应报文的数量。

例 9.2 假设有一个 LAN，每 15 分钟轮询所有被管理设备一次（这在当前的 TCP/IP 网络中是典型的），管理报文的处理时间是 50 ms，网络延迟为 1ms（每个分组 1000 字节），没有产生明显的网络拥塞，Δ 大约是 0.202s，则

$$N \leqslant T/\Delta = 15 \times 60/0.202 = 4500$$

即管理站最多可支持 4500 个设备。

例 9.3 在由多个子网组成的广域网中，网络延迟更大，数据速率更小，通信距离更远，而且还有路由器和网桥引入的延迟，总的网络延迟可能达到半秒钟，Δ 大约是 1.2s，于是有

$$N \leqslant T/\Delta = 15 \times 60/1.2 = 750$$

管理站可支持的设备最多为 750 个。

PDU 类型	请求标识	0	0	变量绑定表

（a）GetRequest、GetNextRequest、SetRequest、InformRequest 和 Trap PDU

PDU 类型	请求标识	错误状态	错误索引	变量绑定表

（b）Response PDU

PDU 类型	请求标识	非重复数 N	最大后继数 M	变量绑定表

（c）GetBulkRequest PDU

变量名 1	值 1	变量名 2	值 2	…	变量名 *n*	值 *n*

（d）变量绑定表

图 9-23　SNMPv2 PDU 格式

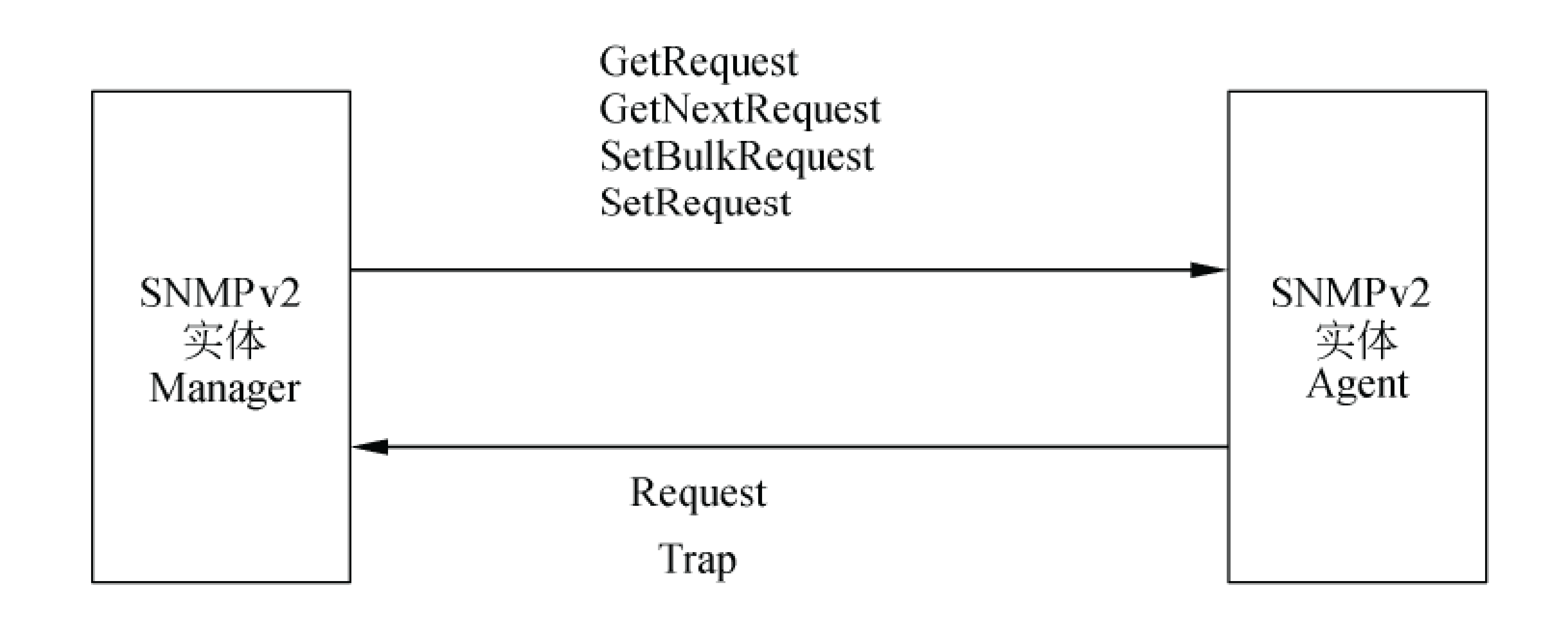


图 9-24　管理站和代理之间的通信

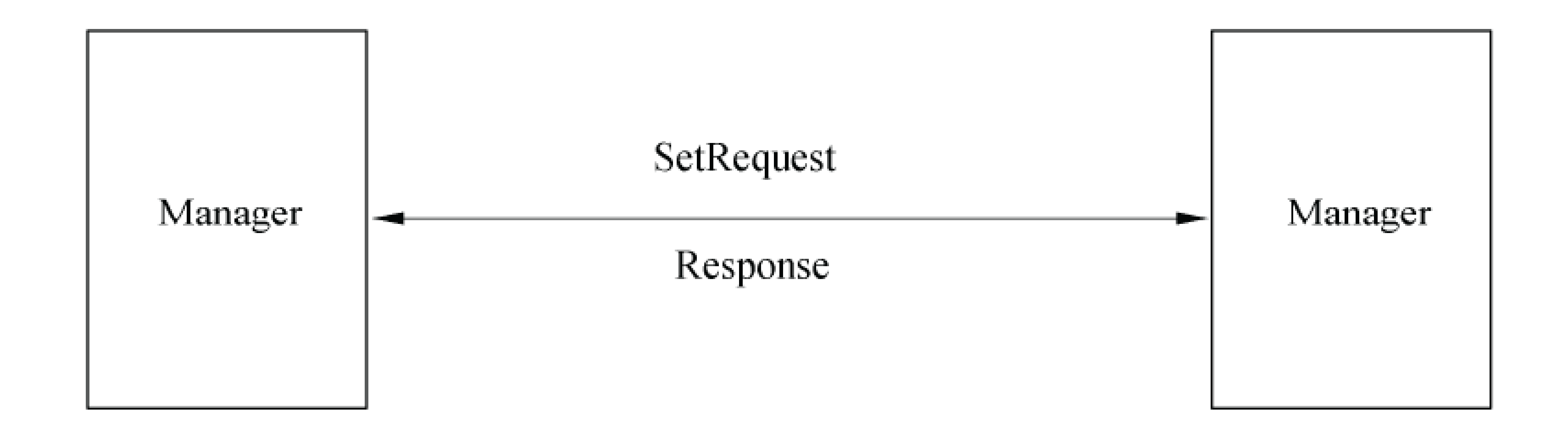
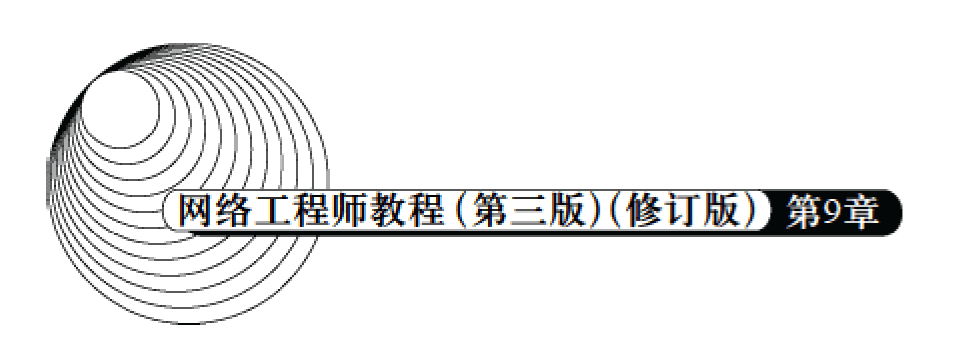


图 9-25　管理站和管理站之间的通信

1．GetRequestPDU

Get 操作用于检索管理信息库中的变量，一次可以检索多个变量的值。接收 GetRequest 的 SNMP 实体以请求标识符相同的 GetResponse 报文响应。在 SNMPv1 中，GetResponse 操作具



有原子性，即只要有一个变量的值检索不到，就不返回任何值。SNMPv2 的响应方式与 SNMPv1 不同，SNMPv2 允许部分响应。如果由于任何其他原因而处理失败，则返回一个错误状态 genErr，对应的错误索引指向有问题的变量。如果生成的响应 PDU 太大，超过了本地的或请求方的最大报文限制，则放弃这个 PDU，构造一个新的响应 PDU，其错误状态为 tooBig，错误索引为 0，变量绑定表为空。

改变 Get 响应的原子性是一个重大进步。在 SNMPv1 中，如果 Get 操作的一个或多个变量不存在，代理就返回错误 noSuchName，剩下的事情完全由管理站处理：要么不向上层返回值；要么去掉不存在的变量，重发检索请求，然后向上层返回部分结果。由于生成部分检索算法的复杂性，很多管理站并不支持这一功能。

2．GetNextRequestPDU

GetNext 命令检索变量名指示的下一个对象实例，用在对表对象的搜索中。在 SNMPv2 中，这种检索请求的格式和语义与 SNMPv1 基本相同，唯一的差别就是改变了响应的原子性。

3．GetBulkRequestPDU

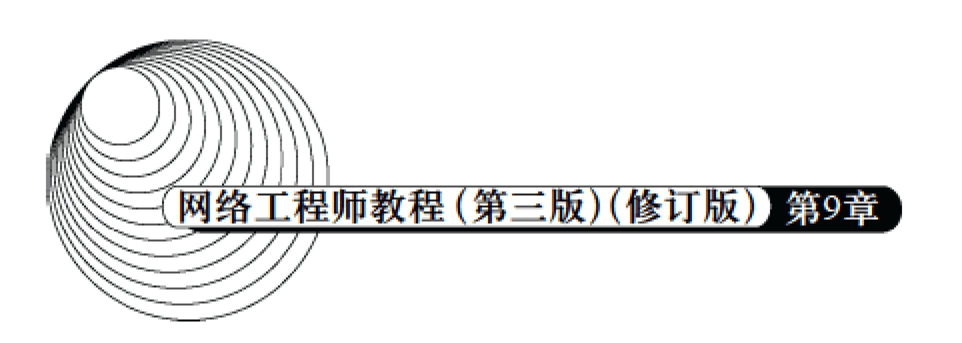
这是 SNMPv2 对原标准的主要增强，目的是以最少的交换次数检索最大量的管理信息。这种块检索操作的工作过程是这样的：假设 GetBulkRequestPDU 变量绑定表中有 L 个变量，GetBulk PDU 的“非重复数”字段的值为 N，则对前 N 个变量应各返回一个后继值。再设 GetBulk PDU 的“最大后继数”字段的值为 M，则对其余的 $R=L-N$ 个变量应该各返回最多 M 个后继值。如果可能，总共返回 $N+R\times M$ 个值，这些值的分布如图 9-26 所示。如果在任何一步查找过程中遇到不存在后继的情况，则返回错误状态 endOfMibView。

4．SetRequestPDU

这个请求 PDU 的格式和语义与 SNMPv1 的基本相同，其语义是设置或改变 MIB 变量的值，其差别是处理响应的方式不同。SNMPv2 实体分两个阶段处理这个请求的变量绑定表，首先是检验操作的合法性，然后再更新变量。如果至少有一个变量绑定对的合法性检验没有通过，则不进行下一阶段的更新操作。所以这个操作与 SNMPv1 一样，是原子性的。如果没有检查出错误，就可以给所有指定变量赋予新值。若有至少一个赋值操作失败，则所有赋值被撤销，并返回错误状态 commitFailed，错误索引指向问题变量的序号。但是，若不能全部撤销所赋的值，则返回错误状态 undoFailed，错误索引字段置 0。

5．TrapPDU

陷入是由代理发给管理站的非确认性消息。SNMPv2 的陷入采用与 Get 等操作相同的 PDU



在前两版中叫做管理站和代理的东西在 SNMPv3 中统一叫做 SNMP 实体（SNMP entity）。实体是体系结构的一种实现，由一个或多个 SNMP 引擎（SNMP engine）和一个或者多个 SNMP 应用（SNMP Application）组成，图 9-27 显示了 SNMP 实体的组成元素。

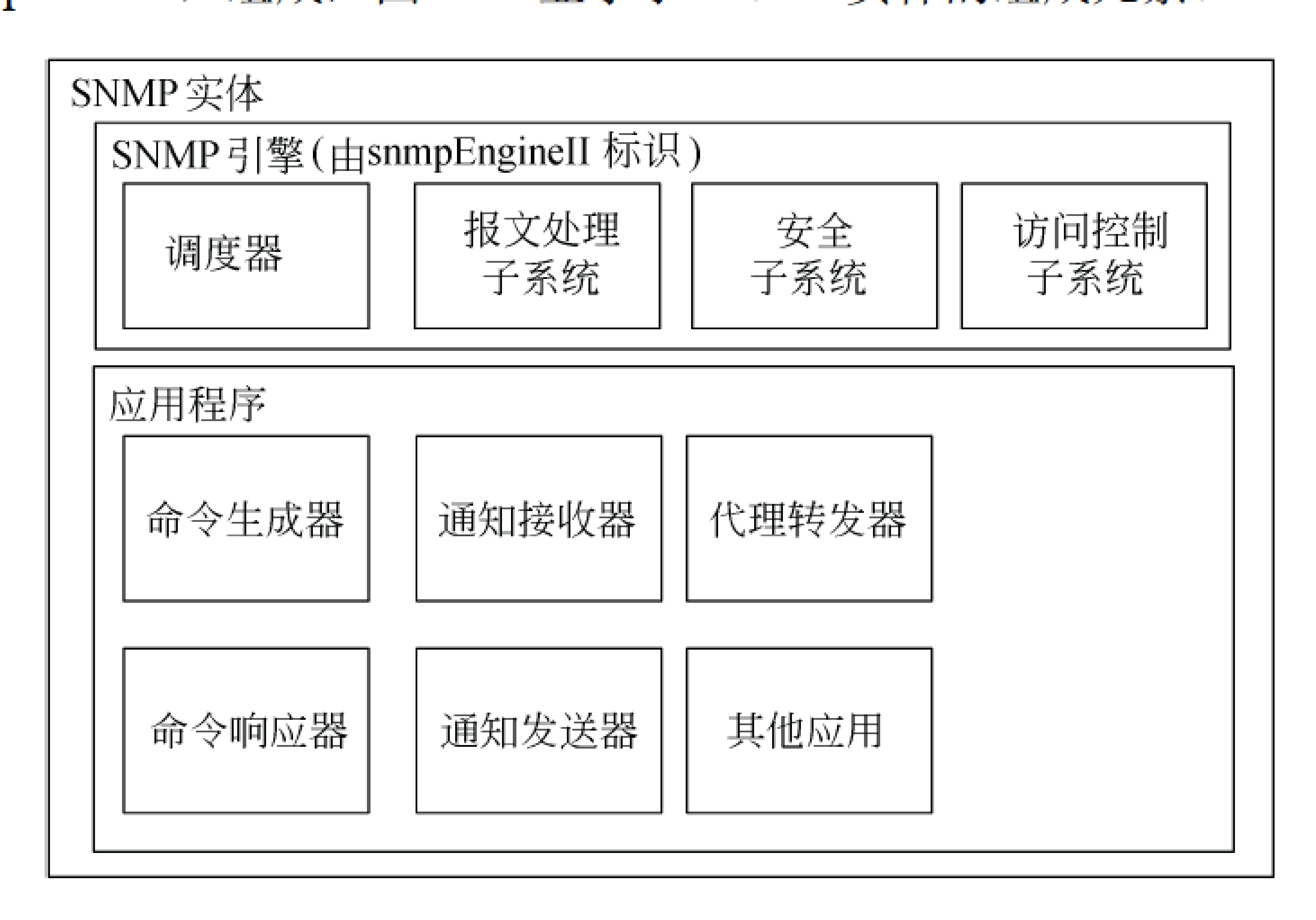


图 9-27　SNMP 实体

1．SNMP 引擎

SNMP 引擎提供下列服务。

（1）发送和接收报文。

（2）认证和加密报文。

（3）控制对管理对象的访问。

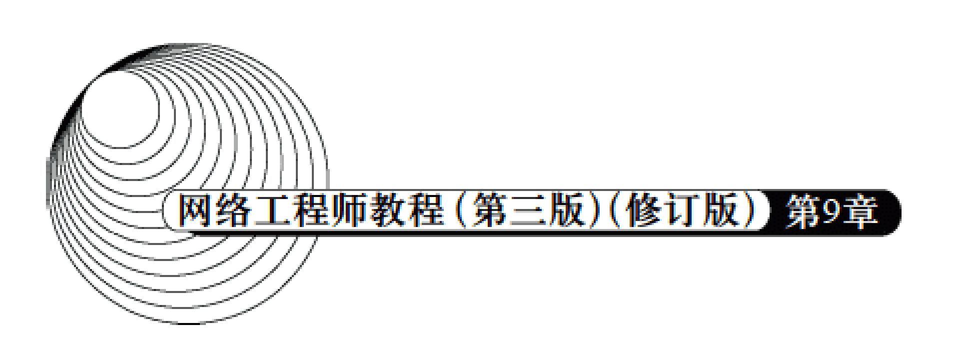
SNMP 引擎有唯一的标识 snmpEngineID，由于 SNMP 引擎和 SNMP 实体具有一一对应的关系，所以 snmpEngineID 也是对应的 SNMP 实体的唯一标识。SNMP 引擎具有复杂的结构，它包含如下部分。

（1）一个调度器（Dispatcher），其作用是发送/接收 SNMP 报文。

（2）一个报文处理子系统（Message Processing Subsystem），其功能是按照预定的格式准备要发送的报文，或者从接收的报文中提取数据。

（3）一个安全子系统（Security Subsystem），提供安全服务，例如报文的认证和加密。一个安全子系统可以有多个安全模块，以便提供各种不同的安全服务。

（4）一个访问控制子系统（Access Control Subsystem），提供授权服务，即确定是否允许访问一个管理对象，或者是否可以对某个管理对象实施特殊的管理操作。



用于共享密钥的两个实体之间，使用散列函数作为密码，所以也叫做 HMAC。HMAC 可以结合任何重复加密的散列函数，例如 MD5 和 SHA-1。可见，HMAC-MD5-96 认证协议就是使用散列函数 MD5 的报文认证协议。

- 加密模块。防止报文内容的泄露。数据的加密使用 DES 算法，使用 56 位的密钥，按照 CBC（Cipher Block Chaining）模式对 64 位长的明文进行替代和替换，最后产生的密文也被分成 64 位的块。

另外，SNMPv3 还对用户密钥进行了局部化处理。用户通常使用可读的 ASCII 字符串作为口令字，密钥局部化就是把用户的口令字变换成他/她与一个 SNMP 引擎共享的密钥。虽然用户在整个网络中可能只使用一个口令，但是通过密钥局部化以后，用户与每一个 SNMP 引擎共享的密钥都是不同的。这样的设计可以防止一个密钥值的泄露对其他 SNMP 引擎造成危害。密钥局部化过程的主要思想是把口令字和相应的 SNMP 引擎标识作为输入，运行一个散列函数（例如 MD5 或 SHA），得到一个固定长度的伪随机序列，作为加密密钥。

4. 基于视图的访问控制（VACM）模型

当一个 SNMP 实体处理检索或修改请求时都要检查是否允许访问指定的管理对象，以及是否允许执行请求的操作。另外，当 SNMP 实体生成通知报文时，也要用到访问控制机制，以决定把消息发送给谁。在 VACM 模型中要用到以下概念。

- SNMP 上下文（context）：简称上下文，是 SNMP 实体可以访问的管理信息的集合。一个管理信息可以存在于多个上下文中，而一个 SNMP 实体也可以访问多个上下文。在一个管理域中，SNMP 上下文由唯一的名字 contextName 标识。
- 组（group）：由二元组<securityModel，securityName>的集合构成。属于同一组的所有安全名 securityName 在指定的安全模型 securityModel 下的访问权限相同。组的名字用 groupName 表示。
- 安全模型（securityModel）：表示访问控制中使用的安全模型。
- 安全级别（securityLevel）：在同一组中成员可以有不同的安全级别，即 noAuthNoPriv（无认证不保密）、authNoPriv（有认证不保密）和 authPriv（有认证要保密）。任何一个访问请求都有相应的安全级别。
- 操作（operation）：指对管理信息执行的操作，例如读、写和发送通知等。

9.5 管理数据库 MIB-2

9.5.1 被管理对象的定义

SNMP 环境中的所有被管理对象组织成树型结构，如图 9-28 和图 9-29 所示。这种层次树

为 SNMP 的实验和改进提供了非常灵活的管理机制。

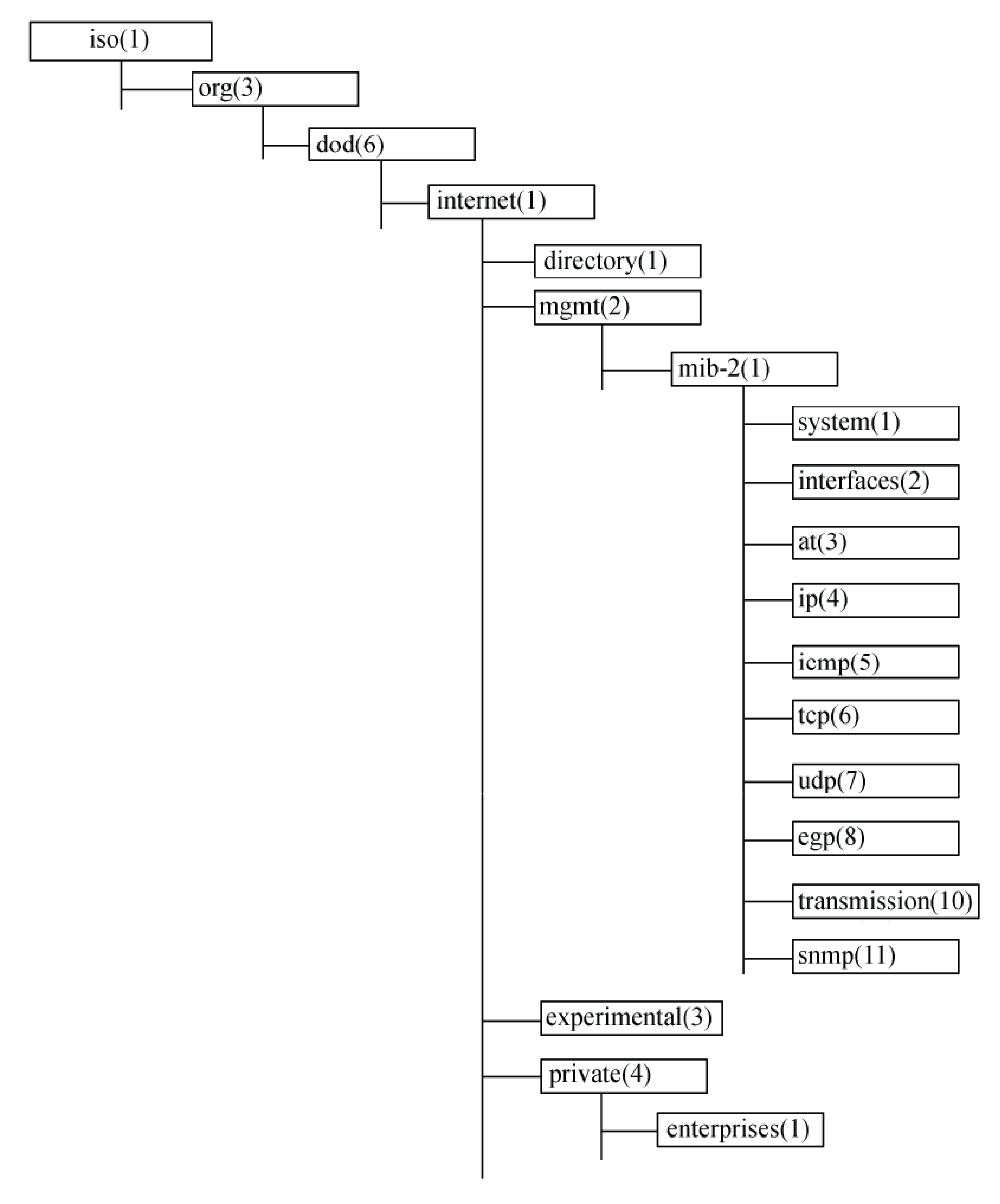


图 9-29　MIB-2 的分组结构

SNMP MIB 中的每个对象属于一定的对象类型，并且有一个具体的值。对象类型的定义采用 ASN.1 描述，对象实例是对象类型的具体实现，只有实例才可以绑定到特定的值。

SNMP MIB 的宏定义最初在 RFC1155 中说明，叫做 MIB-1。后来对 RFC1212 进行了扩充，叫做 MIB-2。图 9-30 是 RFC1212 中对象类型的定义，对其中关键的成分解释如下。

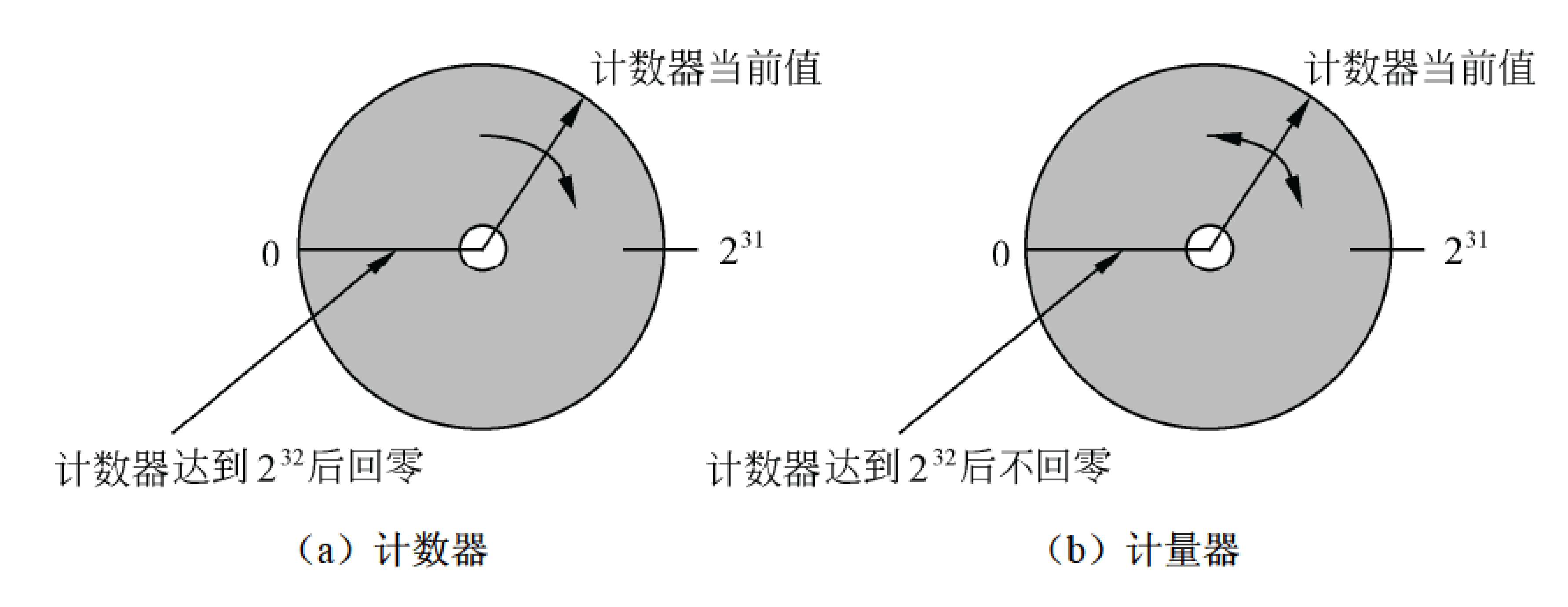
- SYNTAX：语法子句说明被管理对象的类型、它的组成和值的范围，以及与其他对象的关系。对象类型的定义是一种语法描述，对象实例是对象类型的具体实现，只有实例才可以绑定到特定的值。MIB 中使用了 ASN.1 中的 5 种通用类型，如表 9-2 所示。

的类型。如果一个对象被说明为可取消的（deprecated），则表示当前必须支持这种对象，但在将来的标准中可能被取消。

- DesctPart：这个子句是任选的，用文字说明对象类型的含义。
- ReferPart：这个子句也是任选的，用文字说明可参考在其他 MIB 模块中定义的对象。
- IndexPart：用于定义表对象的索引项。
- DefValPart：这个子句是任选的，定义了对象实例默认值。
- VALUE NOTATION：指明对象的访问名。

另外，RFC1155 文件还根据网络管理的需要定义了下列应用类型。

- NetworkAddress：可以有多种网络地址，但目前定义的只有 IP 地址。
- IpAddress：32 位的 IP 地址，定义为 4 个字节的串。
- Counter：计数器类型是一个非负整数，其值可增加，但不能减少，达到最大值 $2^{32}-1$ 后回 0，再从头开始增加，如图 9-31（a）所示。计数器可用于计算接收到的分组数或字节数等。
- Gauge：计量器类型是一个非负整数，其值可增加，也可减少。计量器的最大值也是 $2^{32}-1$。与计数器不同的地方是计量器达到最大值后不回 0，而是锁定在 $2^{32}-1$，如图 9-31（b）所示。计量器可用于表示存储在缓冲队列中的分组数。



（a）计数器　　（b）计量器

图 9-31　计数器和计量器

- TimeTicks：时钟类型是非负整数。时钟的单位是百万分之一秒，可表示从某个事件（例如设备启动）开始到目前经过的时间。
- Opaque：不透明类型，即未知数据类型，可以表示任意类型。这种数据在编码时按字符串处理，管理站和代理都能解释这种类型。

SNMPv2 增加了两种新的数据类型 Unsigned32 和 Counter64。Unsigned32 与 Gauge32 都是 32 位的整数，但是在 SNMPv2 中赋予了不同的语义。Counter64 与 Counter32 一样，都表示计数器，只能增加，不能减少。当增加到 $2^{64}-1$ 或 $2^{32}-1$ 时回 0，从头再增加。而且 SNMPv2 规定，计数器没有定义的初始值，所以计数器的单个值是没有意义的，只有连续两次读计数器得到的

增加值才是有意义的。

SNMPv2 规范澄清了原来标准中一些含糊不清的地方。首先是在 SNMPv2 中规定 Gauge32 的最大值可以设置为小于 2^{32} 的任意正数 MAX，而在 SNMPv1 中 Gauge32 最大值总是 $2^{32}-1$。显然，这样规定更细致了，使用更方便了。其次是 SNMPv2 明确了当计量器达到最大值时可自动减少。而在 RFC1155 中只是说计量器的值“锁定”在最大值，但是锁定的含义并没有定义，人们总是在“计量器达到最大值时是否可以减少”的问题上争论不休。

9.5.2 MIB-2 的功能组

RFC1213 定义了 MIB-2，包含 11 个功能组，共 171 个对象。下面解释主要的功能组。

（1）系统组（System group）。提供了系统的一般信息。表 9-3 所示是系统组的对象。

表 9-3 系统组对象

对　　象	语　　法	访问方式	功 能 描 述
sysDescr（1）	DisplayString（SIZE（0..255））	RO	有关硬件和操作系统的描述
sysObjectID（2）	OBJECT IDENTIFIER	RO	系统制造商标识
sysUpTime（3）	Timeticks	RO	系统运行时间
sysContact（4）	DisplayString（SIZE（0..255））	RW	系统管理人员描述
sysName（5）	DisplayString（SIZE（0..255））	RW	系统名
sysLocation（6）	DisplayString（SIZE（0..255））	RW	系统的物理位置
sysServices（7）	INTEGER（0..127）	RO	系统服务

（2）Interface 组。接口组包含关于主机接口的配置信息和统计信息，如表 9-4 所示。

表 9-4 接口组对象

对　　象	语　　法	访问方式	功 能 描 述
ifNumber	INTEGER	RO	网络接口数
ifTable	SEQUENCE OF ifEntry	NA	接口表
ifEntry	SEQUENCE	NA	接口表项
ifIndex	INTEGER	RO	唯一的索引
ifDescr	DisplayString（SIZE（0..255））	RO	接口描述信息、制造商名、产品名和版本等
ifType	INTEGER	RO	物理层和数据链路层协议确定的接口类型
ifMtu	INTEGER	RO	最大协议数据单元大小（位组数）

续表

对　　象	语　　法	访问方式	功 能 描 述
ifSpeed	Gauge	RO	接口数据速率
ifPhysAddress	PhysAddress	RO	接口物理地址
ifAdminStatus	INTEGER	RW	管理状态 up（1）down（2）testing（3）
ifOperStatus	INTEGER	RO	操作状态 up（1）down（2） testing（3）
ifLastChange	TimeTicks	RO	接口进入当前状态的时间
ifInOctets	Counter	RO	接口收到的总字节数
ifInUcastPkts	Counter	RO	输入的单点传送分组数
ifInNUcastPkts	Counter	RO	输入的组播分组数
ifInDiscards	Counter	RO	丢弃的分组数
ifInErrors	Counter	RO	接收的错误分组数
ifInUnknownPorotos	Counter	RO	未知协议的分组数
ifOutOctets	Counter	RO	通过接口输出的分组数
ifOutUcastPkts	Counter	RO	输出的单点传送分组数
ifOutNUcastPkts	Counter	RO	输出的组播分组数
ifOutDiscards	Counter	RO	丢弃的分组数
ifOutErrors	Counter	RO	输出的错误分组数
ifOutQLen	Gauge	RO	输出队列长度
ifSpecfic	OBJECT IDENTIFIER	RO	指向 MIB 中专用的定义

接口组中的对象可用于故障管理和性能管理。例如，可以通过检查进出接口的字节数或队列长度检测网络拥塞；可以通过接口状态获知工作情况；还可以统计出输入输出的错误率。

输入错误率=ifInErrors/（ifInUcastPkts+ifInNUcastPkts）

输出错误率=ifOutErrors/（ifOutUcastPkts+ifOutNUcastPkts）

另外，该组可以提供接口发送的字节数和分组数，这些数据可作为计费的依据。

（3）地址转换组。地址转换组包含一个表，该表的一行对应系统的一个物理接口，表示网络地址到接口的物理地址的映像关系。MIB-2 中地址转换组的对象已被收编到各个网络协议组中，保留地址转换组仅仅是为了与 MIB-1 兼容。

（4）IP 组。IP 组提供了与 IP 协议有关的信息。由于端系统（主机）和中间系统（路由器）都实现 IP 协议，而这两种系统中包含的 IP 对象又不完全相同，所以有些对象是任选的，这取决于是否与系统有关。IP 组包含的对象如表 9-5 所示。

表 9-5　IP 组对象

对　　象	语　　法	访问方式	功 能 描 述
ipForwarding（1）	INTEGER	RW	IP gateway（1），IP host（2）
ipDefaultTTL（2）	INTEGER	RW	IP 头中的 Time To Live 字段的值
ipInReceives（3）	Counter	RO	IP 层从下层接收的数据报总数
ipInHdrErrors（4）	Counter	RO	由于 IP 头出错而丢弃的数据报
ipInAddrErrors（5）	Counter	RO	地址出错（无效地址、不支持的地址和非本地主机地址）的数据报
ipForwDatagrams（6）	Counter	RO	已转发的数据报
ipInUnknownProtos（7）	Counter	RO	不支持数据报的协议，因而被丢弃
ipInDiscards（8）	Counter	RO	因缺乏缓冲资源而丢弃的数据报
ipInDelivers（9）	Counter	RO	由 IP 层提交给上层的数据报
ipOutRequests（10）	Counter	RO	由 IP 层交给下层需要发送的数据报，不包括 ipForwDatagrams
ipOutDiscards（11）	Counter	RO	在输出端因缺乏缓冲资源而丢弃的数据报
ipOutNoRoutes（12）	Counter	RO	没有到达目标的路由而丢弃的数据报
ipReasmTimeout（13）	INTEGER	RO	数据段等待重装配的最长时间（秒）
ipReasmReqds（14）	Counter	RO	需要重装配的数据段
ipReasmOKs（15）	Counter	RO	成功重装配的数据段
ipReasmFails（16）	Counter	RO	不能重装配的数据段
ipFragOKs（17）	Counter	RO	分段成功的数据段
ipFragFails（18）	Counter	RO	不能分段的数据段
ipFragCreates（19）	Counter	RO	产生的数据报分段数
ipAddrTable（20）	SEQUENCE OF	NA	IP 地址表
ipRouteTable（21）	SEQUENCE OF	NA	IP 路由表
ipNetToMediaTable（22）	SEQUENCE OF	NA	IP 地址转换表
ipRoutingDiscards（23）	Counter	RO	无效的路由项，包括为释放缓冲空间而丢弃路由项

（5）ICMP 组。ICMP 是 IP 的伴随协议。所有实现 IP 协议的节点都必须实现 ICMP 协议。icmp 组包含有关 ICMP 实现和操作的有关信息，它是各种接收的或发送的 ICMP 报文的计数器，如表 9-6 所示。

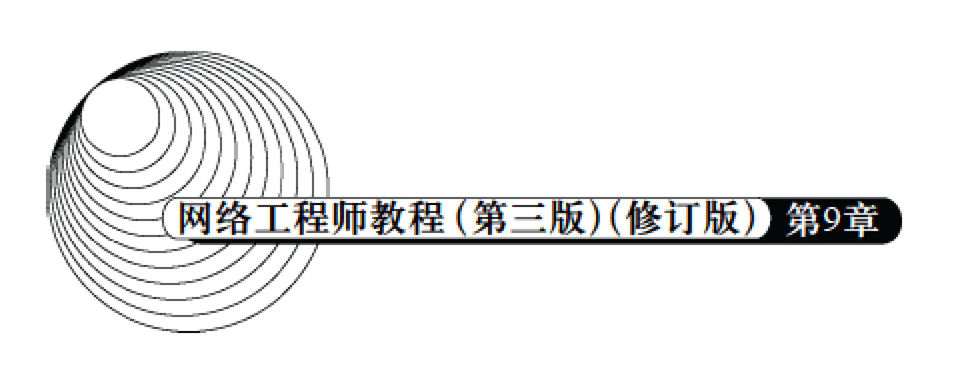


表 9-6　IP 组对象

对　　象	语　　法	访问方式	功 能 描 述
icmpInMsgs（1）	Counter	RO	接收的 icmp 报文总数（以下为输入报文）
icmpInErrors（2）	Counter	RO	出错的 icmp 报文数
icmpInDestUnreachs（3）	Counter	RO	目标不可送达型 icmp 报文
icmpInTimeExcds（4）	Counter	RO	超时型 icmp 报文
icmpInPramProbe（5）	Counter	RO	有参数问题型 icmp 报文
icmpInSrcQuenchs（6）	Counter	RO	源抑制型 icmp 报文
icmpInRedirects（7）	Counter	RO	重定向型 icmp 报文
icmpInEchos（8）	Counter	RO	回声请求型 icmp 报文
icmpInEchoReps（9）	Counter	RO	回声响应型 icmp 报文
icmpInTimestamps（10）	Counter	RO	时间戳请求型 icmp 报文
icmpInTimestampReps（11）	Counter	RO	时间戳响应型 icmp 报文
icmpInAddrMasks（12）	Counter	RO	地址掩码请求型 icmp 报文
icmpInAddrMaskReps（13）	Counter	RO	地址掩码响应型 icmp 报文
icmpOutMsgs（14）	Counter	RO	输出的 icmp 报文总数（以下为输出报文）
icmpOutErrors（15）	Counter	RO	出错的 icmp 报文数
icmpOutDestUnreachs（16）	Counter	RO	目标不可送达型 icmp 报文
icmpOutTimeExcds（17）	Counter	RO	超时型 icmp 报文
icmpOutPramProbe（18）	Counter	RO	有参数问题型 icmp 报文
icmpOutSrcQuenchs（19）	Counter	RO	源抑制型 icmp 报文
icmpOutRedirects（20）	Counter	RO	重定向型 icmp 报文
icmpOutEchos（21）	Counter	RO	回声请求型 icmp 报文
icmpOutEchoReps（22）	Counter	RO	回声响应型 icmp 报文
icmpOutTimestamps（23）	Counter	RO	时间戳请求型 icmp 报文
icmpOutTimestampReps（24）	Counter	RO	时间戳响应型 icmp 报文
icmpOutAddrMasks（25）	Counter	RO	地址掩码请求型 icmp 报文
icmpOutAddrMaskReps（26）	Counter	RO	地址掩码响应型 icmp 报文

（6）TCP 组。TCP 组包含与 TCP 协议的实现和操作有关的信息，这一组的前三项与重传有关。当一个 TCP 实体发送数据段后就等待应答，并开始计时。如果超时后没有得到应答，就认为数据段丢失了，因而要重新发送。TCP 组包含的对象如表 9-7 所示。

表 9-7 TCP 组对象

对　　象	语　　法	访问方式	功 能 描 述
tcpRtoAlgorithm（1）	INTEGER	RO	重传时间算法
tcpRtoMin（2）	INTEGER	RO	重传时间最小值
tcpRtoMax（3）	INTEGER	RO	重传时间最大值
tcpMaxConn（4）	INTEGER	RO	可建立的最大连接数
tcpActiveOpens（5）	Counter	RO	主动打开的连接数
tcpPassiveOpens（6）	Counter	RO	被动打开的连接数
tcpAttemptFails（7）	Counter	RO	连接建立失败数
tcpEstabResets（8）	Counter	RO	连接复位数
tcpCurrEstab（9）	Gauge	RO	状态为 established 或 closeWait 的连接数
tcpInSegs（10）	Counter	RO	接收的 TCP 段总数
tcpOutSegs（11）	Counter	RO	发送的 TCP 段总数
tcpRetransSegs（12）	Counter	RO	重传的 TCP 段总数
tcpConnTable（13）	SEQUENCE OF	NA	连接表
tcpInErrors（14）	Counter	RO	接收的出错 TCP 段数
tcpOutRests（15）	Counter	RO	发出的含 RST 标志的段数

（7）UDP 组。UDP 组类似于 TCP 组，它包含了关于 UDP 数据报和本地接收端点的详细信息。

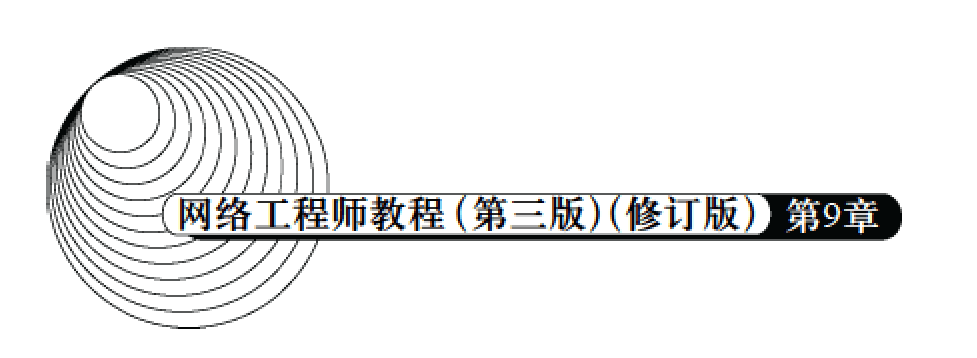
（8）EGP 组。EGP 组提供了关于 EGP 路由器发送和接收的 EGP 报文的信息，以及关于 EGP 邻居的详细信息等。

（9）传输组。设置这一组的目的是针对各种传输介质提供详细的管理信息，事实上这不是一个组，而是一个联系各种接口专用信息的特殊节点。前面介绍过的接口组包含各种接口通用的信息，而传输组则提供与子网类型有关的专用信息。

9.5.3 SNMPv2 管理信息库

SNMPv2 MIB 扩展和细化了 MIB-2 中定义的管理对象，又增加了新的管理对象。

（1）系统组。SNMPv2 的系统组是 MIB-2 系统组的扩展，图 9-32 表示出这个组的管理对象。可以看出，这个组只是增加了与对象资源（Object Resource）有关的一个标量对象 sysORLastChange 和一个表对象 sysORTable，它仍然属于 MIB-2 的层次结构。所谓对象资源，是指由代理实体使用和控制的、可以由管理站动态配置的系统资源。标量对象 sysORLastChange 记录着对象资源表中描述的对象实例改变状态（或值）的时间。对象资源表是一个只读的表，每一个可动态配置的对象资源占用一个表项。



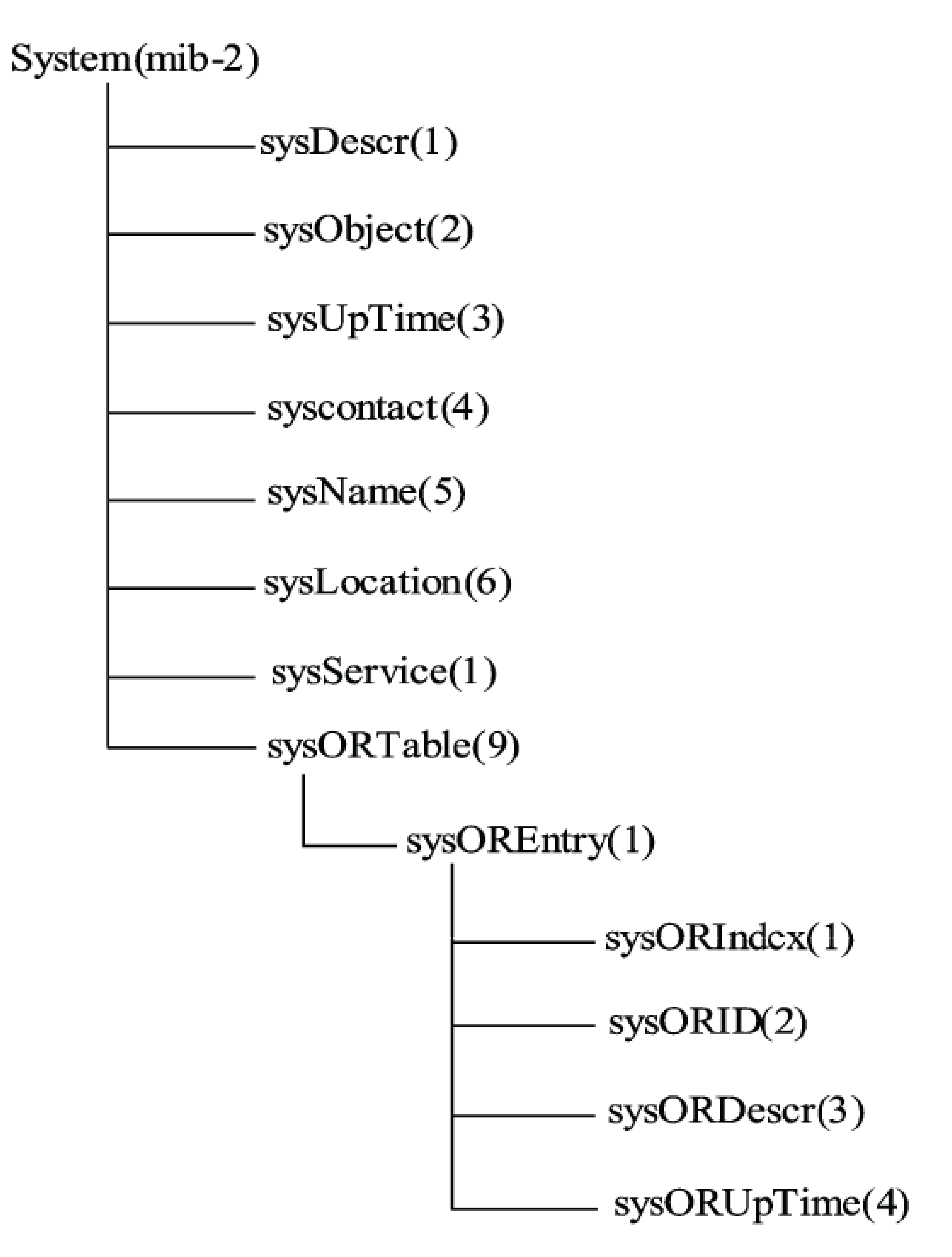


图 9-32　SNMPv2 系统组

（2）SNMP 组。这个组是由 MIB-2 的对应组改造而成的，有些对象被删除了，同时又增加了一些新对象，如图 9-33 所示。

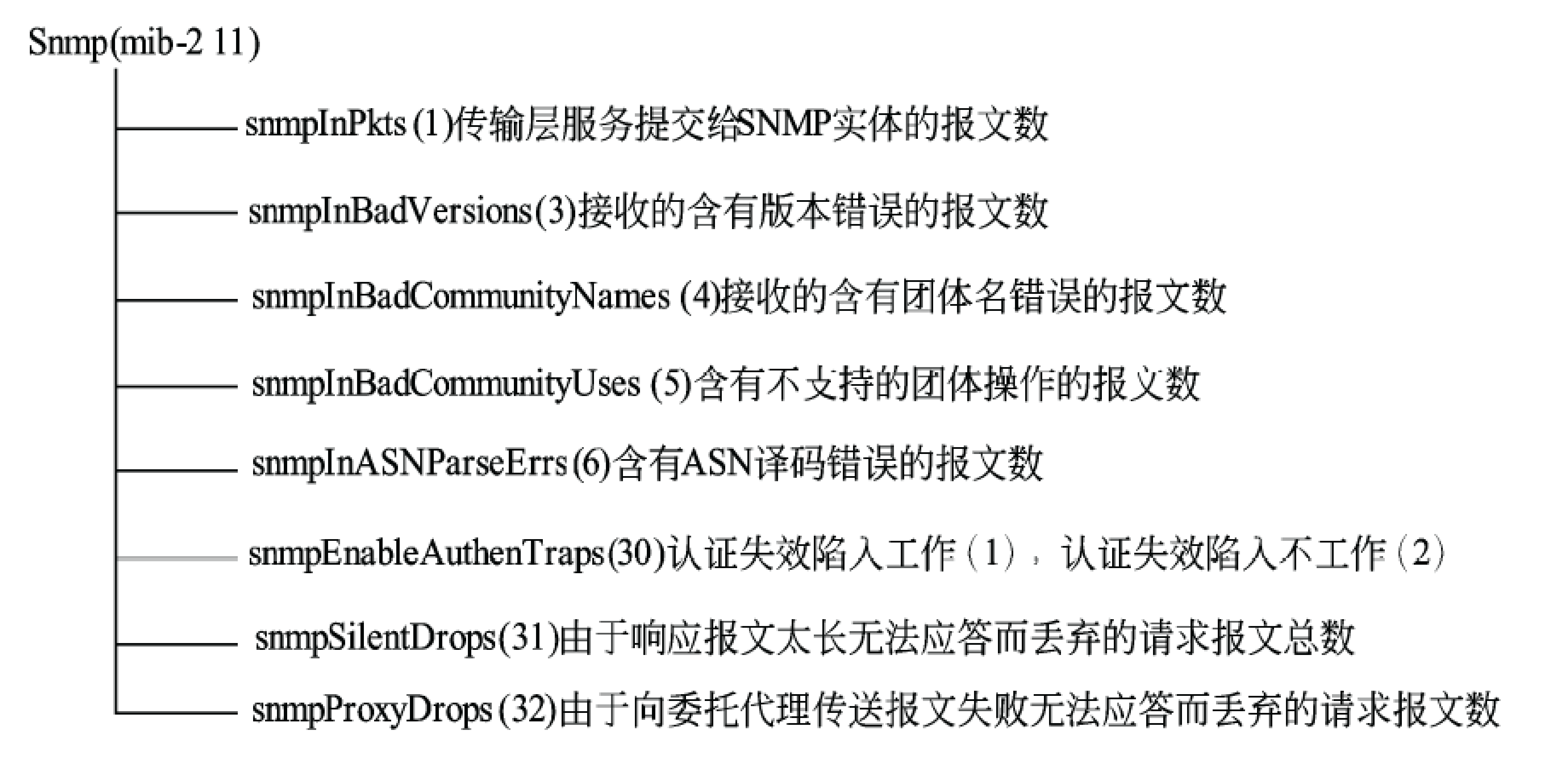


图 9-33　改进的 SNMP 组

（3）MIB 对象组。这个新组包含的对象与管理对象的控制有关，分为两个子组，如图 9-34 所示。第一个子组 snmpTrap 由两个对象组成。

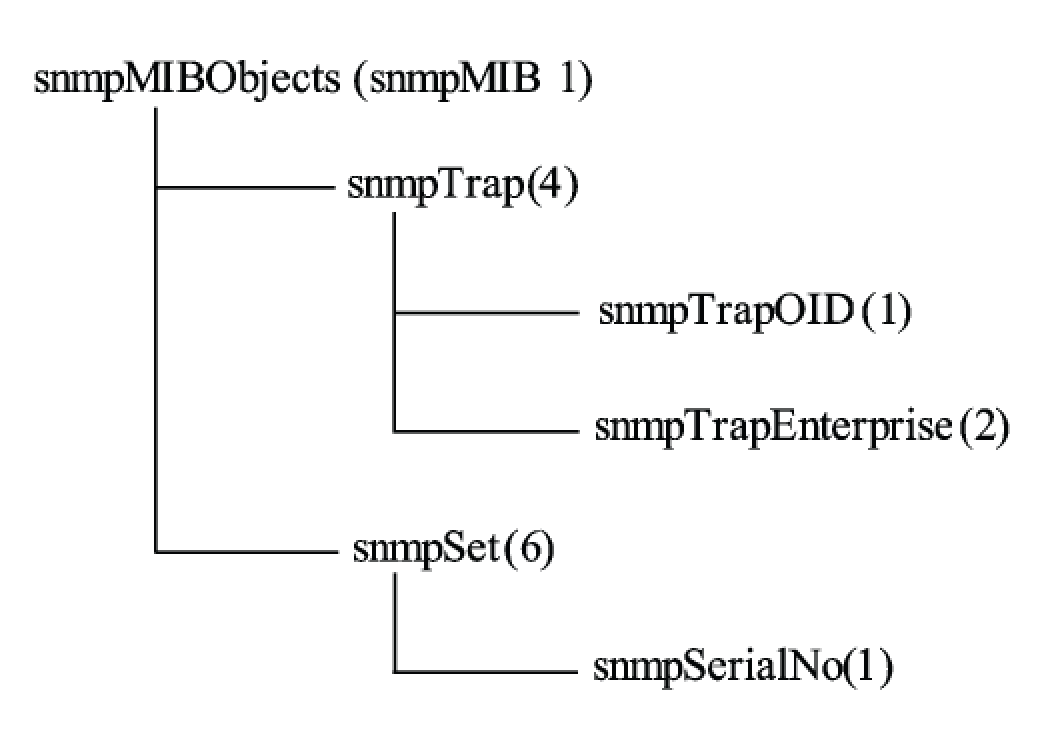


图 9-34　SNMP MIB 对象组

- snmpTrapOID：这是正在发送的陷入或通知的对象标识符，这个变量出现在陷入 PDU 或通知请求 PDU 的变量绑定表中的第二项。
- snmpTrapEnterprise：这是与正在发送的陷入有关的制造商的对象标识符，当 SNMPv2 的委托代理把一个 RFC1157 陷入 PDU 映像到 SNMPv2 陷入 PDU 时，这个变量出现在变量绑定表的最后。

第二个子组 snmpSet 仅有一个对象 snmpSerialNo，这个对象用于解决 set 操作中可能出现的两个问题。

① 一个管理站可能向同一 MIB 对象发送多个 set 操作，保证这些操作按照发送的顺序在 MIB 中执行是必要的，即使在传送过程中次序发生了错乱也是这样。

② 多个管理站对 MIB 的并发操作可能破坏了数据库的一致性和精确性。

（4）接口组。MIB-2 定义的接口组经过一段时间的使用，发现有很多缺陷。RFC1573 分析了原来的接口组没有提供的功能和其他不足之处。

① 接口编号。MIB-2 接口组定义变量 ifNumber 作为接口编号，而且是常数，这对于允许动态增加/删除网络接口的协议（例如 SLIP/PPP）是不合适的。

② 接口子层。有时需要区分网络层下面的各个子层，而 MIB-2 没有提供这个功能。

③ 虚电路问题。对应一个网络接口可能有多个虚电路。

④ 不同传输特性的接口。MIB-2 接口表记录的内容只适合基于分组传输的协议，不适合面向字符的协议（例如 PPP，EIA RS-232），也不适合面向位的协议（例如 DS1）和固定信息长度传输的协议（例如 ATM）。

⑤ 计数长度。当网络速度增加时，32 位的计数器经常溢出回 0。

⑥ 接口速度。ifSpeed 最大为（2^{32}–1）bps，但是现在有的网络速度已远远超过这个限制，例如 SONET OC-48 为 2.448Gbps。

⑦ 组播/广播分组计数。MIB-2 接口组不区分组播分组和广播分组，但分别计数有时是有用的。

⑧ 接口类型。ifType 表示接口类型，MIB-2 定义的接口类型不能动态增加，只能在推出新的 MIB 版本时再增加，而这个过程一般需要几年时间。

⑨ ifSpecific 问题。MIB-2 对这个变量的定义很含糊。有的实现给这个变量赋予介质专用的 MIB 的对象标识符，而有的实现赋予介质专用表的对象标识符，或者是这种表的入口对象标识符，甚至是表的索引对象标识符。

根据以上分析，RFC1573 对 MIB-2 接口组做了一些小的修改，纠正了上面提到的问题。例如，重新规定 ifIndex 不再代表一个接口，而是用于区分接口子层，而且不再限制 ifIndex 的取值必须在 1～ifNumber 之间。这样对应一个物理接口可以有多个代表不同逻辑子层的表行，还允许动态地增加/删除网络接口。RFC1573 废除了有些用处不大的变量，例如 ifInNUcastPkts 和 ifOutNUPkts，它们的作用已经被接口扩展表中的新变量代替。由于变量 ifOutQLen 在实际中很少实现，也被废除了。变量 ifSpecific 由于前述原因也被废除了，它的作用已被 ifType 代替。同时把 ifType 的语法改变为 IANAifType，而这种类型可以由 Internet 编码机构（Internet Assigned Number Authorty）随时更新，从而不受 MIB 版本的限制。

9.6 RMON

9.6.1 RMON 的基本概念

通常用于监视整个网络通信情况的设备叫做网络监视器（Monitor）或网络分析器（Analyzer）、探测器（Probe）等。监视器观察 LAN 上出现的每个分组，并进行统计和总结，给管理人员提供重要的管理信息。监视器还能存储部分分组，供以后分析用。监视器也根据分组类型进行过滤并捕获特殊的分组。通常是每个子网配置一个监视器，并且与中央管理站通信，因此叫做远程监视器，如图 9-35 所示。图中监视器可以是一个独立设备，也可以是运行监视器软件的工作站或服务器等。中央管理站具有 RMON 管理能力，能够与各个监视器交换管理信息。RMON 监视器或探测器（RMON Probe）实现 RMON 管理信息库（RMON MIB）。这种系统与通常的 SNMP 代理一样包含一般的 MIB，另外还有一个探测器进程，提供与 RMON 有关的功能。探测器进程能够读写本地的 RMON 数据库，并响应管理站的查询请求。所以也把 RMON 探测器称为 RMON 代理。

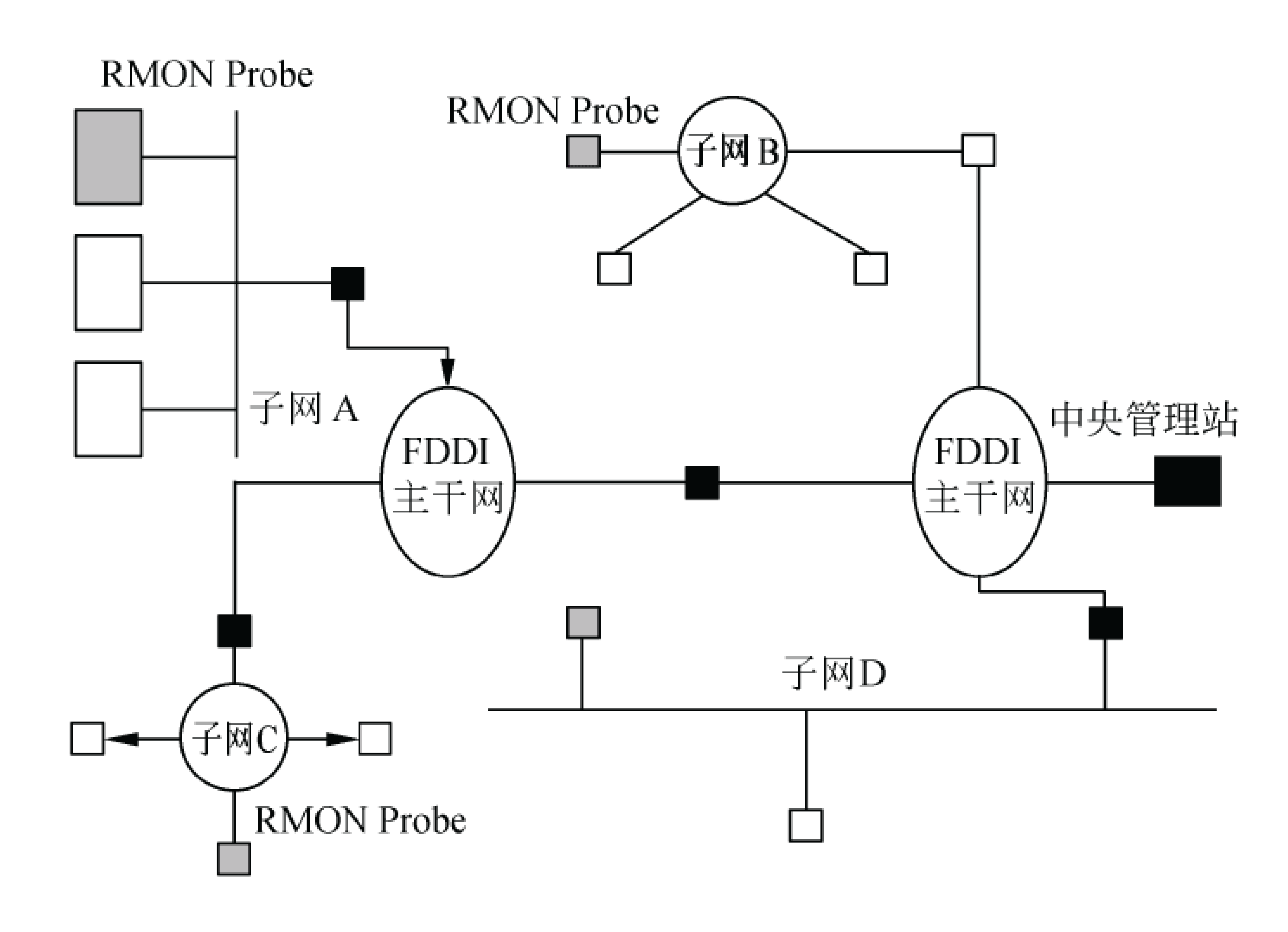


图 9-35　远程网络监视的配置

RMON 定义了远程网络监视的管理信息库，以及 SNMP 管理站与远程监视器之间的接口。一般地说，RMON 的目标就是监视子网范围内的通信，从而减少管理站和被管理系统之间的通信负担。更具体地说，RMON 有下列目标。

（1）离线操作。必要时管理站可以停止对监视器的轮询，有限的轮询可以节省网络带宽和通信费用。

（2）主动监视。如果监视器有足够的资源，通信负载也容许，监视器可以连续地或周期地运行诊断程序，收集并记录网络性能参数。

（3）问题检测和报告。如果主动监视消耗网络资源太多，监视器也可以被动地获取网络数据。

（4）提供增值数据。监控器可以分析收集到的子网数据，从而减轻了管理站的计算任务。

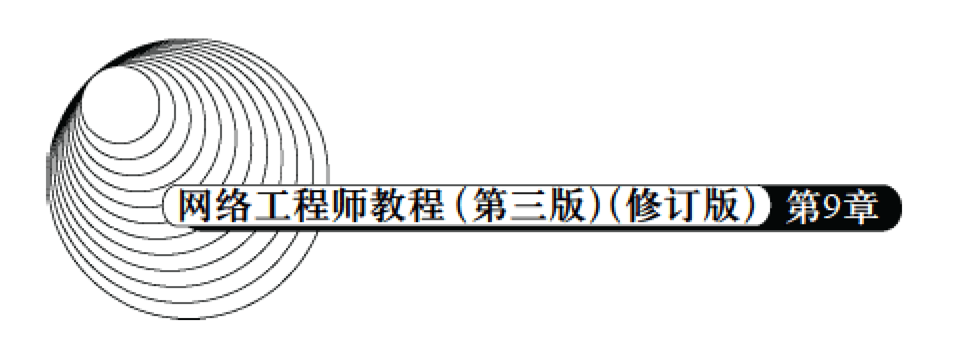
（5）多管理站操作。一个因特网可能有多个管理站，这样可以提高可靠性，或者分布地实现各种不同的管理功能。

9.6.2　RMON 的管理信息库

RMON 规范定义了管理信息库 RMON MIB，它是 MIB-2 下面的第 16 个子树。RMON MIB 分为 10 组，如图 9-36 所示。存储在每一组中的信息都是监视器从一个或几个子网中统计和收集的数据。这 10 个功能组都是任选的，但实现时有下列联带关系。

（1）实现警报组时必须实现事件组，警报就是对某种网络事件的警告。

（2）实现最高 N 台主机组时必须实现主机组，因为最高 N 台主机组是从主机组中提取出



来的。

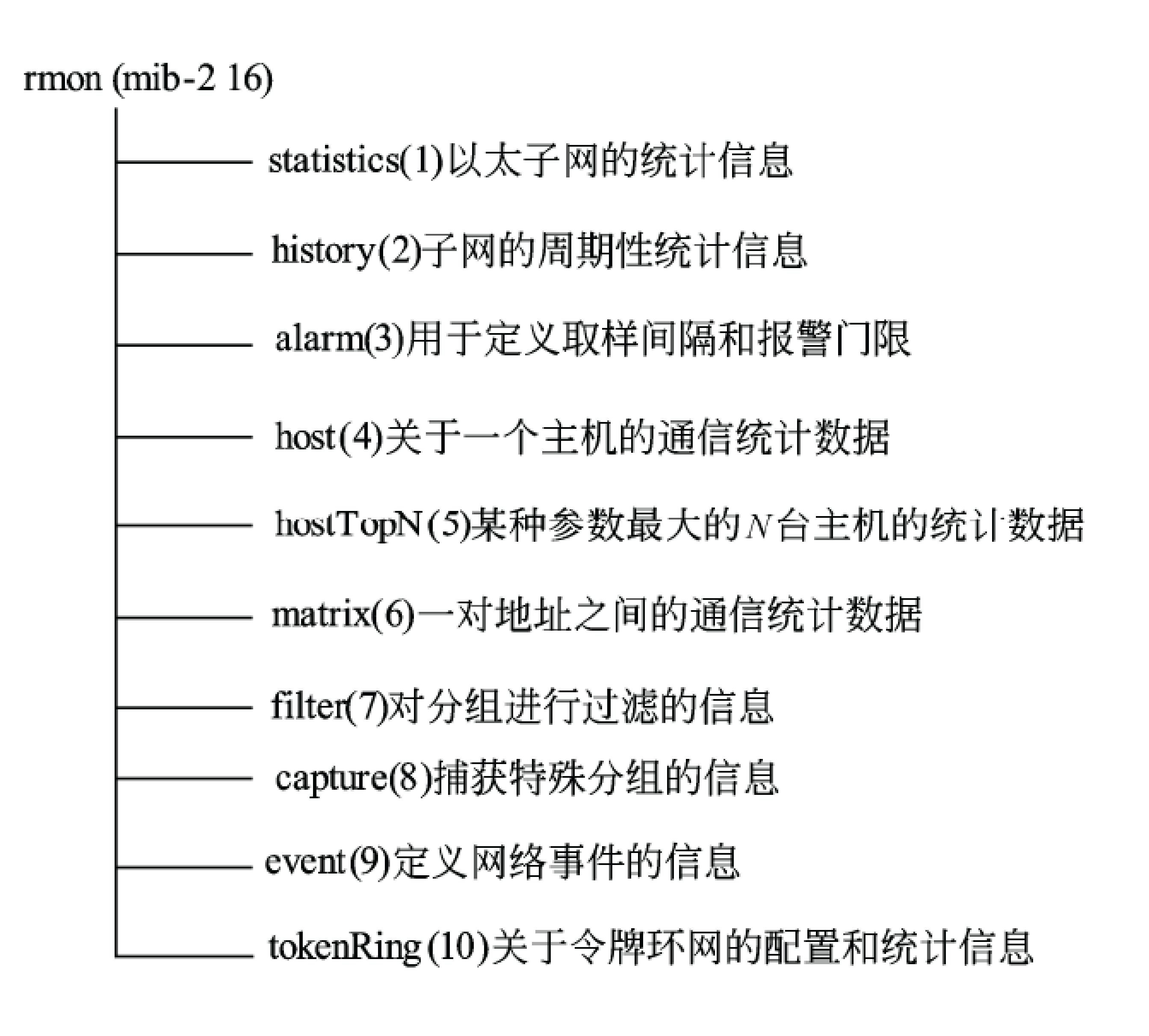


图 9-36　RMON MIB 子树

（3）实现捕获组时必须实现过滤组，经过过滤的分组可以被捕获。

9.6.3　RMON2 的管理信息库

RMON2 监视 OSI/RM 第 3～7 层的通信，能对数据链路层以上的分组进行译码。这使得监视器可以管理包括 IP 协议等网络层协议，因而能了解分组的源和目标地址，能知道路由器负载的来源，使得监视的范围扩大到局域网之外。监视器也能监视应用层协议，例如电子邮件协议、文件传输协议和 HTTP 协议等，这样监视器就可以记录主机应用活动的数据，可以显示各种应用活动的图表。这些对网络管理人员都是很重要的信息。另外，在网络管理标准中，通常把网络层之上的协议都叫做应用层协议，以后提到的应用层包含 OSI 的 5，6，7 层。

RMON2 扩充了原来的 RMON MIB，增加了 9 个新的功能组，如图 9-37 所示。

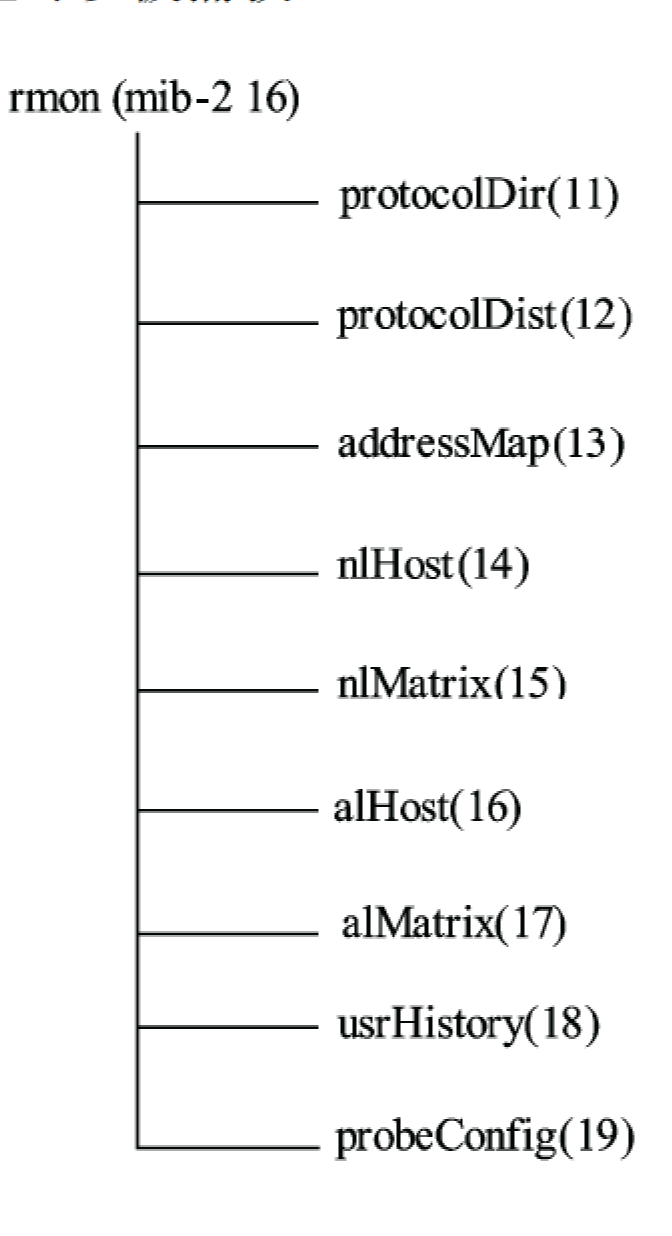


图 9-37　RMON2 MIB

9.7 网络诊断和配置命令

Windows 提供了一组实用程序来实现简单的网络配置和管理功能，这些实用程序通常以 DOS 命令的形式出现。用键盘命令来显示和改变网络配置，感觉就像直接操控硬件一样，不但操作简单方便，而且效果立即显现；不但能详细了解网络的配置参数，而且提高了网络管理的效率。所以掌握常用的网络管理命令是网络管理人员的基本技能，必须坚持使用，才能驾轻就熟。

Windows 的网络管理命令通常以 exe 文件的形式存储在 system32 目录中，在“开始”菜单中运行命令解释程序 Cmd.exe 就进入 DOS 命令窗口，可以执行任何实用程序。下面的一些例子都是在 DOS 窗口中截图的。

9.7.1 Ipconfig

Ipconfig 命令相当于 Windows 9x 中的图形化命令 Winipcfg，是最常用的 Windows 实用程序，可以显示所有网卡的 TCP/IP 配置参数，可以刷新动态主机配置协议（DHCP）和域名系统的设置。Ipconfig 的语法如下：

ipconfig [/all] [/renew[*Adapter*]] [/release[*Adapter*]] [/flushdns] [/displaydns] [/registerdns] [/showclassid *Adapter*] [/setclassid *Adapter* [*ClassID*]]

对以上命令参数解释如下：

- /?

显示帮助信息，对本章中其他命令有同样作用。

- /all

显示所有网卡的 TCP/IP 配置信息。如果没有该参数，则只显示各个网卡的 IP 地址、子网掩码和默认网关地址。

- /renew [*Adapter*]

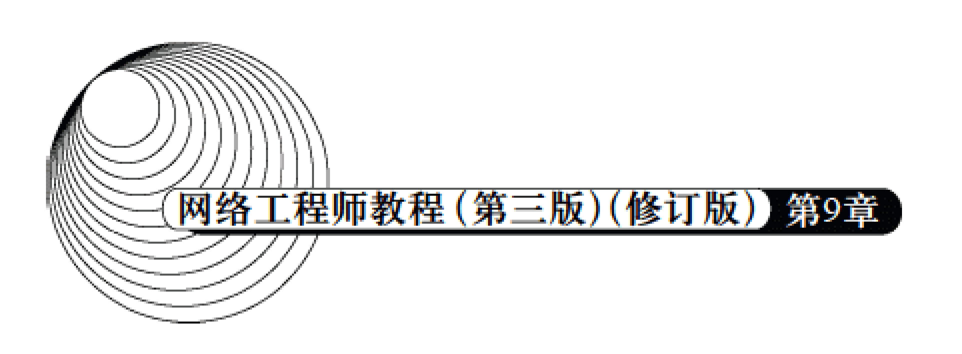
更新网卡的 DHCP 配置，如果使用标识符 *Adapter* 说明了网卡的名字，则只更新指定网卡的配置，否则就更新所有网卡的配置。这个参数只能用于动态配置 IP 的计算机。使用不带参数的 ipconfig 命令，可以列出所有网卡的名字。

- /release[*Adapter*]

向 DHCP 服务器发送 DHCP Release 请求，释放网卡的 DHCP 配置参数和当前使用的 IP 地址。

- /flushdns

刷新客户端 DNS 缓存的内容。在 DNS 排错期间，可以使用这个命令丢弃负缓存项以及其



他动态添加的缓存项。

- /displaydns

显示客户端 DNS 缓存的内容，该缓存中包含从本地主机文件中添加的预装载项，以及最近通过名字解析查询得到的资源记录。DNS 客户端服务使用这些信息快速处理经常出现的名字查询。

- /registerdns

刷新所有 DHCP 租约，重新注册 DNS 名字。在不重启计算机的情况下，可以利用这个参数来排除 DNS 名字注册中的故障，解决客户端和 DNS 服务器之间的手工动态更新问题。可以利用“高级 TCP/IP 设置”来注册本地连接的 DNS 后缀，如图 9-38 所示。

- /showclassid *Adapter*

显示网卡的 DHCP 类别 ID。利用通配符“*”代替标识符 Adapter，可以显示所有网卡的 DHCP 类别 ID。这个参数仅适用于自动配置 IP 地址的计算机。可以根据某种标准把 DHCP 客户端划分成不同的类别，以便于管理。例如，移动客户划分到租约期较短的类，固定客户划分到租约期较长的类。

- /setclassid *Adapter*[*ClassID*]

对指定的网卡设置 DHCP 类别 ID。如果未指定 DHCP 类别 ID，则会删除当前的类别 ID。

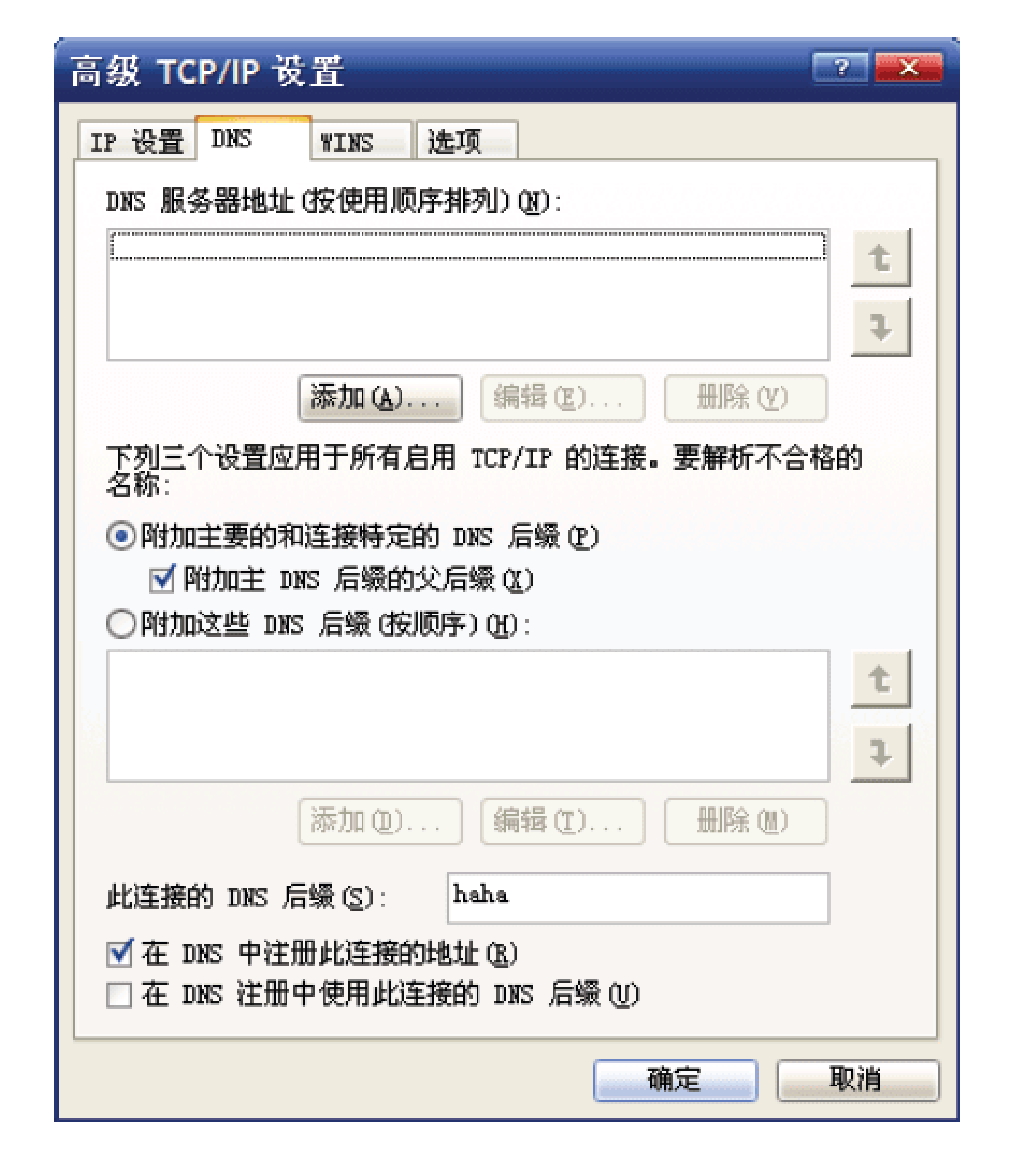


图 9-38　高级 TCP/IP 设置

如果 Adapter 名称包含空格，则要在名称两边使用引号（即 "Adapter 名称"）。网卡名称中可以使用通配符星号“*”，例如，Local*可以代表所有以字符串 Local 开头的网卡，而*Con*可以表示所有包含字符串 Con 的网卡。

ipconfig 命令最适合于自动分配 IP 地址的计算机，使用户可以明确区分 DHCP 或自动专用 IP 地址（APIPA）配置的参数。

举例如下。

（1）如果要显示所有网卡的基本 TCP/IP 配置参数，输入：

```
ipconfig
```

（2）如果要显示所有网卡的完整 TCP/IP 配置参数，输入：

```
ipconfig /all
```

（3）如果仅更新本地连接的网卡由 DHCP 分配的 IP 地址，输入：

ipconfig /renew "Local Area Connection"

（4）在排除 DNS 名称解析故障时，如果要刷新 DNS 解析器缓存，输入：

ipconfig /flushdns

（5）如果要显示名称以 Local 开头的所有网卡的 DHCP 类别 ID，输入：

ipconfig /showclassid Local*

（6）如果要将“本地连接”网卡的 DHCP 类别 ID 设置为 TEST，输入：

ipconfig /setclassid "Local Area Connection" TEST

图 9-39 是用 ipconfig/all 命令显示的网络配置参数，其中列出了主机名、网卡物理地址和 DHCP 租约期，由 DHCP 分配的 IP 地址、子网掩码、默认网关和 DNS 服务器的 IP 地址等配置参数。图 9-40 是利用参数 showclassid 显示的“本地连接”的类别标识。

```
C:\Documents and Settings\Administrator>ipconfig/all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : x4ep512rdszwjzp
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Unknown
   IP Routing Enabled. . . . . . . . : Yes
   WINS Proxy Enabled. . . . . . . . : Yes

Ethernet adapter 本地连接:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : SiS 900-Based PCI Fast Ethernet Adapter
   Physical Address. . . . . . . . . : 00-03-0D-07-03-7F
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IP Address. . . . . . . . . . . . : 100.100.17.24
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 100.100.17.254
   DHCP Server . . . . . . . . . . . : 192.168.254.10
   DNS Servers . . . . . . . . . . . : 218.30.19.40
                                       61.134.1.4
   Lease Obtained. . . . . . . . . . : 2009年1月5日 8:10:14
   Lease Expires . . . . . . . . . . : 2009年1月5日 12:10:14
```

图 9-39　ipconfig 命令显示的结果

```
C:\Documents and Settings\Administrator>ipconfig /showclassid 本地连接

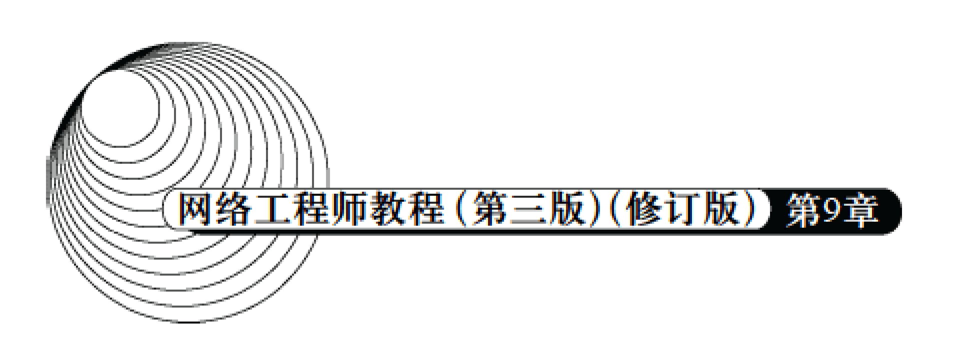
Windows IP Configuration

DHCP Classes for Adapter "本地连接":

   DHCP ClassID Name . . . . . . . . . : 默认路由和远程访问类别
   DHCP ClassID Description  . . . . . : 远程访问客户端的用户类别

   DHCP ClassID Name . . . . . . . . . : 默认 BOOTP 的类别
   DHCP ClassID Description  . . . . . : BOOTP 客户端的用户类别
```

图 9-40　ipconfig/showclassid 命令显示的结果



9.7.2 Ping

Ping 命令通过发送 ICMP 回声请求报文来检验与另外一个计算机的连接。这是一个用于排除连接故障的测试命令，如果不带参数则显示帮助信息。Ping 命令的语法如下：

ping [-t] [-a] [-n *Count*] [-l *Size*] [-f] [-i *TTL*] [-v *TOS*] [-r *Count*] [-s *Count*] [{-j *HostList* | -k *HostList*}] [-w *Timeout*] [*TargetName*]

对以上命令参数解释如下。

- -t

持续发送回声请求直至输入 Ctrl+Break 或 Ctrl+C 被中断，前者显示统计信息，后者不显示统计信息。

- -a

用 IP 地址表示目标，进行反向名字解析，如果命令执行成功，则显示对应的主机名。

- -n *Count*

说明发送回声请求的次数，默认为 4 次。

- -l *Size*

说明了回声请求报文的字节数，默认是 32，最大为 65 527。

- -f

在 IP 头中设置不分段标志，用于测试通路上传输的最大报文长度。

- -i *TTL*

说明 IP 头中 TTL 字段的值，通常取主机的 TTL 值，对于 Windows XP 主机，这个值是 128，最大为 255。

- -v *TOS*

说明了 IP 头中 TOS（Type of Service）字段的值，默认是 0。

- -r *Count*

在 IP 头中添加路由记录选项，Count 表示源和目标之间的跃点数，其值在 1～9 之间。

- -s *Count*

在 IP 头中添加时间戳（timestamp）选项，用于记录达到每一跃点的时间，Count 的值在 1～4 之间。

- -j *HostList*

在 IP 头中使用松散源路由选项，HostList 指明中间节点（路由器）的地址或名字，最多 9 个，用空格分开。

- -k *HostList*

在 IP 头中使用严格源路由选项，HostList 指明中间节点（路由器）的地址或名字，最多 9

个，用空格分开。

- -w *Timeout*

指明等待回声响应的时间（μs），如果响应超时，则显示出错信息 Request timed out，默认超时间隔为 4 s。

- *TargetName*

用 IP 地址或主机名表示目标设备。

使用 Ping 命令必须安装并运行 TCP/IP 协议。可以使用 IP 地址或主机名来表示目标设备。如果 ping 一个 IP 地址成功，而 ping 对应的主机名失败，则可以断定名字解析有问题。无论名字解析是通过 DNS、NetBIOS，还是通过本地主机文件，都可以用这个方法进行故障诊断。

举例如下。

（1）如果要测试目标 10.0.99.221 并进行名字解析，则输入：

ping -a 10.0.99.221

（2）如果要测试目标 10.0.99.221，发送 10 次请求，每个响应为 1000 字节，则输入：

ping -n 10 -l 1000 10.0.99.221

（3）如果要测试目标 10.0.99.221，并记录 4 个跃点的路由，则输入：

ping -r 4 10.0.99.221

（4）如果要测试目标 10.0.99.221，并说明松散源路由，则输入：

ping -j 10.12.0.1 10.29.3.1 10.1.44.1 10.0.99.221

图 9-41 所示为 ping www.163.com.cn的结果。

```
C:\Documents and Settings\Administrator>ping www.163.com.cn

Pinging www.163.com.cn [219.137.167.157] with 32 bytes of data:

Reply from 219.137.167.157: bytes=32 time=29ms TTL=54
Reply from 219.137.167.157: bytes=32 time=29ms TTL=54
Reply from 219.137.167.157: bytes=32 time=29ms TTL=54
Reply from 219.137.167.157: bytes=32 time=29ms TTL=54

Ping statistics for 219.137.167.157:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 29ms, Maximum = 29ms, Average = 29ms
```

图 9-41 ping 命令的显示结果

9.7.3 Arp

Arp 命令用于显示和修改地址解析协议缓存表的内容，缓存表项是 IP 地址与网卡地址对。

计算机上安装的每个网卡各有一个缓存表。如果使用不含参数的 arp 命令，则显示帮助信息。Arp 命令的语法如下：

arp [-a [*InetAddr*] [-N *IfaceAddr*]] [-g [*InetAddr*] [-N *IfaceAddr*]] [-d *InetAddr* [*IfaceAddr*]] [-s *InetAddr* *EtherAddr* [*IfaceAddr*]]

对以上命令参数解释如下。

- -a [*InetAddr*] [-N *IfaceAddr*]

显示所有接口的 ARP 缓存表。如果要显示特定 IP 地址的 ARP 表项，则使用参数 InetAddr；如果要显示指定接口的 ARP 缓存表，则使用参数-N IfaceAddr。这里，N 必须大写。InetAddr 和 IfaceAddr 都是 IP 地址。

- -g [*InetAddr*] [-N *IfaceAddr*]

与参数-a 相同。

- -d *InetAddr* [*IfaceAddr*]

删除由 InetAddr 指示的 ARP 缓存表项。要删除特定接口的 ARP 缓存表项，使用参数 IfaceAddr 指明接口的 IP 地址。要删除所有 ARP 缓存表项，使用通配符“*”代替参数 InetAddr。

- -s *InetAddr* *EtherAddr* [*IfaceAddr*]

添加一个静态的 ARP 表项，把 IP 地址 InetAddr 解析为物理地址 EtherAddr。参数 IfaceAddr 指定了接口的 IP 地址。

IP 地址 InetAddr 和 IfaceAddr 用点分十进制表示。物理地址 EtherAddr 由 6 个字节组成，每个字节用两个十六进制数表示，字节之间用连字符“-”分开，例如 00-AA-00-4F-2A-9C。

用参数-s 添加的 ARP 表项是静态的，不会由于超时而被删除。如果 TCP/IP 协议停止运行，ARP 表项都被删除。为了生成一个固定的静态表项，可以在批文件中加入适当的 ARP 命令，并在机器启动时运行批文件。

举例如下。

（1）要显示 ARP 缓存表的内容，输入：

```
arp -a
```

（2）要显示 IP 地址为 10.0.0.99 的接口的 ARP 缓存表，输入：

```
arp -a -N 10.0.0.99
```

（3）要添加一个静态表项，把 IP 地址 10.0.0.80 解析为物理地址 00-AA-00-4F-2A-9C，则输入：

```
arp -s 10.0.0.80 00-AA-00-4F-2A-9C
```

图 9-42 所示为使用 arp 命令添加一个静态发表项的例子。

```
C:\Documents and Settings\Administrator>arp -a

Interface: 100.100.17.17 --- 0x10003
  Internet Address      Physical Address      Type
  100.100.17.254        00-0f-e2-29-31-c1     dynamic

C:\Documents and Settings\Administrator>arp -s 202.117.17.254 00-1c-4f-52-2a-8c

C:\Documents and Settings\Administrator>arp -a

Interface: 100.100.17.17 --- 0x10003
  Internet Address      Physical Address      Type
  100.100.17.72         00-1e-8c-ad-f9-ce     dynamic
  100.100.17.75         00-40-d0-53-bf-86     dynamic
  100.100.17.254        00-0f-e2-29-31-c1     dynamic
  202.117.17.254        00-1c-4f-52-2a-8c     static
```

图 9-42　使用 arp 命令的例

9.7.4　Netstat

Netstat 命令用于显示 TCP 连接、计算机正在监听的端口、以太网统计信息、IP 路由表、IPv4 统计信息（包括 IP、ICMP、TCP 和 UDP 等协议）和 IPv6 统计信息（包括 IPv6、ICMPv6、TCP over IPv6 和 UDP over IPv6 等协议）等。如果不使用参数，则显示活动的 TCP 连接。Netstat 命令的语法如下：

netstat [-a] [-e] [-n] [-o] [-p *Protocol*] [-r] [-s] [*Interval*]

对以上参数解释如下。

- -a

显示所有活动的 TCP 连接，以及正在监听的 TCP 和 UDP 端口。

- -e

显示以太网统计信息，例如发送和接收的字节数，以及出错的次数等。这个参数可以与-s 参数联合使用。

- -n

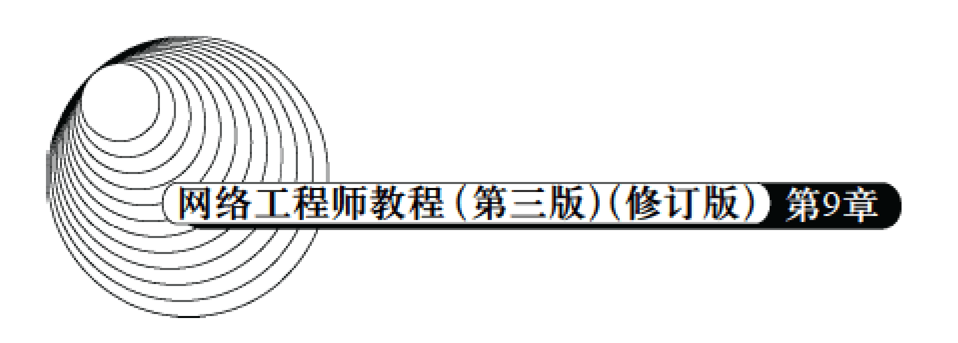
显示活动的 TCP 连接，地址和端口号以数字形式表示。

- -o

显示活动的 TCP 连接以及每个连接对应的进程 ID。在 Windows 任务管理器中可以找到与进程 ID 对应的应用。这个参数可以与-a、-n 和-p 联合使用。

- -p *Protocol*

用标识符 Protocol 指定要显示的协议，可以是 TCP、UDP、TCPv6 或者 UDPv6。如果与参数-s 联合使用，则可以显示协议 TCP、UDP、ICMP、IP、TCPv6、UDPv6、ICMPv6 或 IPv6 的



统计数据。

- -r

显示 IP 路由表的内容，其作用等价于路由打印命令 route print。

- -s

显示每个协议的统计数据。默认情况下，统计 TCP、UDP、ICMP 和 IP 协议发送及接收的数据包、出错的数据包、连接成功或失败的次数等。如果与-p 参数联合使用，可以指定要显示统计数据的协议。

- *Interval*

说明重新显示信息的时间间隔，输入 Ctrl+C 则停止显示。如果不使用这个参数，则只显示一次。

Netstat 显示的统计信息分为 4 栏或 5 栏，解释如下。

- Proto：表示协议的名字（例如 TCP 或 UDP）。
- Local Address：本地计算机的地址和端口。通常显示本地计算机的名字和端口名字（例如 ftp），如果使用了-n 参数，则显示本地计算机的 IP 地址和端口号。如果端口尚未建立，则用“*”表示。
- Foreign Address：远程计算机的地址和端口。通常显示远程计算机的名字和端口名字（例如 ftp），如果使用了-n 参数，则显示远程计算机的 IP 地址和端口号。如果端口尚未建立，则用“*”表示。
- State：表示 TCP 连接的状态，用下面的状态名字表示。
    - ◆ CLOSE_WAIT：收到对方的连接释放请求。
    - ◆ CLOSED：连接已关闭。
    - ◆ ESTABLISHED：连接已建立。
    - ◆ FIN_WAIT_1：已发出连接释放请求。
    - ◆ FIN_WAIT_2：等待对方的连接释放请求。
    - ◆ LAST_ACK：等待对方的连接释放应答。
    - ◆ LISTEN：正在监听端口。
    - ◆ SYN_RECEIVED：收到对方的连接建立请求。
    - ◆ SYN_SEND：已主动发出连接建立请求。
    - ◆ TIMED_WAIT：等待一段时间后将释放连接。

举例如下。

（1）要显示以太网的统计信息和所有协议的统计信息，则输入：

```
netstat -e -s
```

（2）要显示 TCP 和 UDP 协议的统计信息，则输入：

netstat -s -p tcp udp

（3）要显示 TCP 连接及其对应的进程 ID，每 4 s 显示一次，则输入：

nbtstat -o 4

（4）要以数字形式显示 TCP 连接及其对应的进程 ID，则输入：

nbtstat -n -o

图 9-43 是命令 netstat –o 4 显示的统计信息，每 4 s 显示一次，直到输入 Ctrl+C 结束。

```
C:\Documents and Settings\Administrator>netstat  -o 4

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    x4ep512rdszwjzp:1172   121.11.159.208:http    SYN_SENT        1572

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    x4ep512rdszwjzp:1173   121.11.159.208:http    SYN_SENT        1572

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    x4ep512rdszwjzp:1173   121.11.159.208:http    SYN_SENT        1572

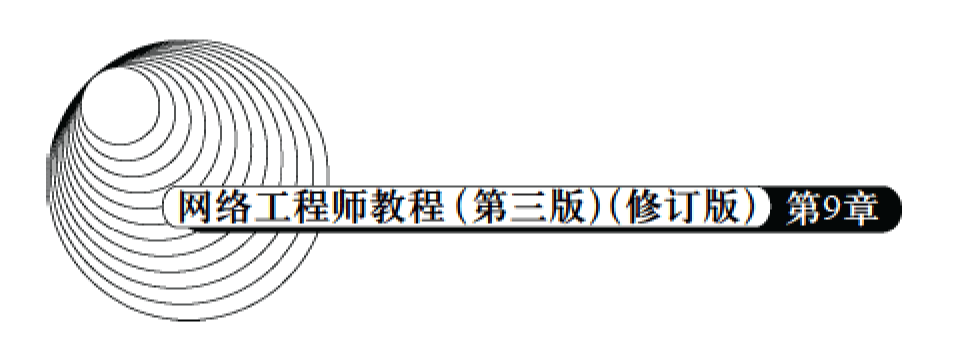
Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    x4ep512rdszwjzp:1176   124.115.3.126:http     ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1178   124.115.6.52:http      ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1179   124.115.6.52:http      ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1180   124.115.6.52:http      ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1182   124.115.3.126:http     ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1183   124.115.6.52:http      ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1184   124.115.6.52:http      ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1185   222.73.73.173:http     ESTABLISHED     3096
  TCP    x4ep512rdszwjzp:1186   222.73.78.14:http      SYN_SENT        3096
```

图 9-43　命令 netstat –o 4 显示的统计信息

9.7.5　Tracert

Tracert 命令的功能是确定到达目标的路径，并显示通路上每一个中间路由器的 IP 地址。通过多次向目标发送 ICMP 回声（echo）请求报文，每次增加 IP 头中 TTL 字段的值，就可以确定到达各个路由器的时间。显示的地址是路由器接近源这一边的端口地址。Tracert 命令的语法如下：




```
tracert [-d] [-h MaximumHops] [-j HostList] [-w Timeout] [TargetName]
```

对以上参数解释如下。

- -d

不进行名字解析，显示中间节点的 IP 地址，这样可以加快跟踪的速度。

- -h *MaximumHops*

说明地址搜索的最大跃点数，默认值是 30 跳。

- -j *HostList*

说明发送回声请求报文要使用 IP 头中的松散源路由选项，标识符 HostList 列出必须经过的中间节点的地址或名字，最多可以列出 9 个中间节点，各个中间节点用空格隔开。

- -w *Timeout*

说明了等待 ICMP 回声响应报文的时间（μs），如果接收超时，则显示星号“*”，默认超时间隔是 4 s。

- *TargetName*

用 IP 地址或主机名表示的目标。

这个诊断工具通过多次发送 ICMP 回声请求报文来确定到达目标的路径，每个报文中 TTL 字段的值都是不同的。通路上的路由器在转发 IP 数据报之前先要对 TTL 字段减一，如果 TTL 为 0，则路由器就向源端返回一个超时（Time Exceeded）报文，并丢弃原来要转发的报文。在 tracert 第一次发送的回声请求报文中置 TTL=1，然后每次加 1，这样就能收到沿途各个路由器返回的超时报文，直至收到目标返回的 ICMP 回声响应报文。如果有的路由器不返回超时报文，那么这个路由器就是不可见的，显示列表中用星号“*”表示。

举例如下。

（1）要跟踪到达主机 corp7.microsoft.com 的路径，则输入：

```
tracert corp7.microsoft.com
```

（2）要跟踪到达主机 corp7.microsoft.com 的路径，并且不进行名字解析，只显示中间节点的 IP 地址，则输入：

```
tracert -d corp7.microsoft.com
```

（3）要跟踪到达主机 corp7.microsoft.com 的路径，并使用松散源路由，则输入：

```
tracert -j 10.12.0.1 10.29.3.1 10.1.44.1 corp7.microsoft.com
```

图 9-44 所示为利用命令 tracert www.163.com.cn显示的路由跟踪列表。

```
C:\Documents and Settings\Administrator>tracert www.163.com.cn

Tracing route to www.163.com.cn [219.137.167.157]
over a maximum of 30 hops:

  1    26 ms    15 ms    11 ms  100.100.17.254
  2    <1 ms    <1 ms    <1 ms  254-20-168-128.cos.it-comm.net [128.168.20.254]

  3    <1 ms    <1 ms    <1 ms  61.150.43.65
  4    <1 ms    <1 ms    <1 ms  222.91.155.5
  5    <1 ms    <1 ms    <1 ms  125.76.189.81
  6     1 ms    <1 ms    <1 ms  61.134.0.13
  7    28 ms    28 ms    28 ms  202.97.35.229
  8    28 ms    29 ms    29 ms  61.144.3.17
  9    29 ms    29 ms    32 ms  61.144.5.9
 10    32 ms    32 ms    32 ms  219.137.11.53
 11    29 ms    29 ms    28 ms  219.137.167.157

Trace complete.
```

图 9-44　tracert 的显示结果

9.7.6　Pathping

Pathping 结合了 ping 和 tracert 两个命令的功能，可以显示通信线路上每个子网的延迟和丢包率。pathping 在一段时间内向通路中的各个路由器发送多个回声请求报文，然后根据每个路由器返回的数据包计算统计结果。由于 pathping 命令显示了每个路由器（或链路）丢失数据包的程度，所以用户可以据此确定哪些路由器或者子网存在通信问题。Pathping 命令的语法如下：

pathping [-n] [-h *MaximumHops*] [-g *HostList*] [-p *Period*] [-q *NumQueries* [-w *Timeout*] [-T] [-R] [*TargetName*]

对以上参数解释如下。

- -n

不进行名字解析，以加快显示速度。

- -h *MaximumHops*

说明了搜索目标期间的最大跃点数，默认是 30。

- -g *HostList*

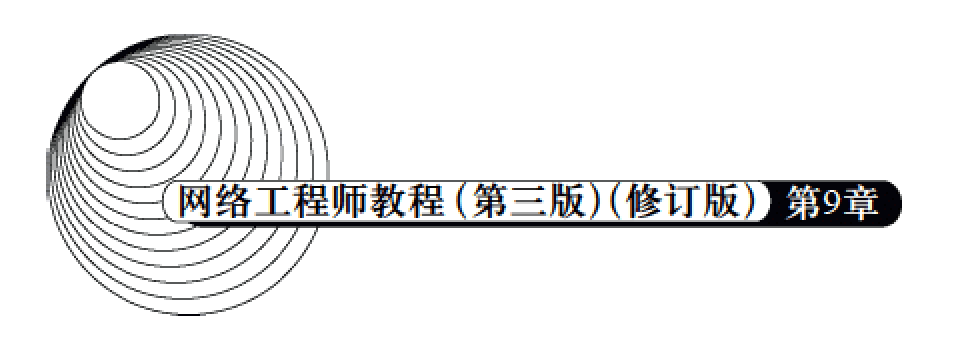
在发送回声请求报文时使用松散源路由，标识符 HostList 列出了中间节点的名字或地址。最多可以列出 9 个中间节点，用空格分开。

- -p *Period*

说明两次 ping 之间的时间间隔（ms），默认为 1/4 s。

- -q *NumQueries*

说明发送给每个路由器的回声请求报文的数量，默认为 100 个。



- -w *Timeout*

说明每次等待回声响应的时间，默认是 3 s。

- -T

对发送的回声请求数据包附加上第二层优先标志（例如 802.1p）。这样可以测试出不具备区分第二层优先级能力的设备，这个开关用于测试网络连接提供不同服务质量的能力。

- -R

确定通路上的设备是否支持资源预约协议（RSVP），这个开关用于测试网络连接提供不同服务质量的能力。

- *TargetName*

用 IP 地址或名字表示的目标。

pathping 命令的参数是大小写敏感的，所以 T 和 R 必须大写。为了防止网络拥塞，ping 的频率不能太快，这样也可以防止突发性地丢包。

当使用-p *Period* 参数时，对每一个中间节点一次只发送一个回声请求包，对同一节点，两次 ping 之间的时间间隔是 Period×跃点数。

当使用-w *Timeout* 参数时，多个回声请求包并行地发出，因此标识符 Timeout 规定的时间并不受由 Period 规定的时间限制。

IEEE 802.1p 标准使得局域网交换机具有以优先级区分信息流的能力，向支持声音、图像和数据的综合业务方面迈进了一步。802.1p 定义了 8 种不同的优先级，分别用于支持时间关键的通信（例如 RIP 和 OSPF 的路由更新报文），延迟敏感的应用（例如交互式语音和视频），可控负载的多媒体流，重要的 SAP 数据以及尽力而为（best-effort）的通信等。符合 802.1p 规范的交换机具有多队列缓冲硬件，可以对较高优先级的分组进行快速处理，使得这些分组能够越过低级别分组而迅速通过交换机。

在传统的单一缓冲区交换机中，当信息传输出现拥塞时，所有分组将平等地排队等待，直到可继续前进。由于传统设备不能识别第二层优先级标签，那些带有优先标签的分组就会被丢弃，所以应用开关 T 可以区分传统交换机与可提供第二层优先级的交换机。

R 参数用于对资源预约协议的测试。RSVP 预约报文在会话开始之前首先发送给通路上的每一个设备。如果设备不支持 RSVP，它返回一个 ICMP“目标不可到达”报文；如果设备支持 RSVP，它返回一个“预约错误信息”报文。有一些设备什么信息也不返回，如果这种情况出现，则显示超时信息。

图 9-45 的例子显示了命令 C:\>pathping -n corp1 的输出。pathping 运行时产生的第一个结果就是路径列表，与 tracert 命令显示的结果相同。接着出现一个大约 125 s 的“忙”消息，忙时间的长短随着跃点数的多少有所变化。这期间，从上述列表中的路由器以及它们之间的链路

收集统计信息，最后显示测试结果。

```
Tracing route to corp1 [10.54.1.196]
over a maximum of 30 hops:
  0  172.16.87.35
  1  172.16.87.218
  2  192.168.52.1
  3  192.168.80.1
  4  10.54.247.14
  5  10.54.1.196
Computing statistics for 125 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           172.16.87.35
                                0/ 100 =  0%   |
  1   41ms     0/ 100 =  0%     0/ 100 =  0%  172.16.87.218
                               13/ 100 = 13%   |
  2   22ms    16/ 100 = 16%     3/ 100 =  3%  192.168.52.1
                                0/ 100 =  0%   |
  3   24ms    13/ 100 = 13%     0/ 100 =  0%  192.168.80.1
                                0/ 100 =  0%   |
  4   21ms    14/ 100 = 14%     1/ 100 =  1%  10.54.247.14
                                0/ 100 =  0%   |
  5   24ms    13/ 100 = 13%     0/ 100 =  0%  10.54.1.196
Trace complete.
```

图 9-45　命令 pathping 的显示结果

在图 9-45 所示的样本报告中，Node/Link、Lost/Sent=Pct 和 Address 栏显示：在 172.16.87.218 与 192.168.52.1 之间的链路上丢包率是 13%。第二跳和第四跳的路由器也丢失了数据包，但是对于它们转发的通信量不会产生影响。在图中的地址栏（Address）中，以直杠“|”标志由于链路拥塞而产生的丢包，至于路由器丢包的原因，则可能是设备过载了。

9.7.7　Nbtstat

这个命令显示 NetBT（NetBIOS over TCP/IP）协议的统计信息，包括本地计算机和远程计算机的 NetBIOS 名字表，以及 NetBIOS 名字缓存。Nbtstat 也可以刷新 NetBIOS 名字缓存，刷新已经注册了的 WINS 名字。Nbtstat 命令的语法如下：

nbtstat [-a *RemoteName*] [-A *IPAddress*] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [*Interval*]

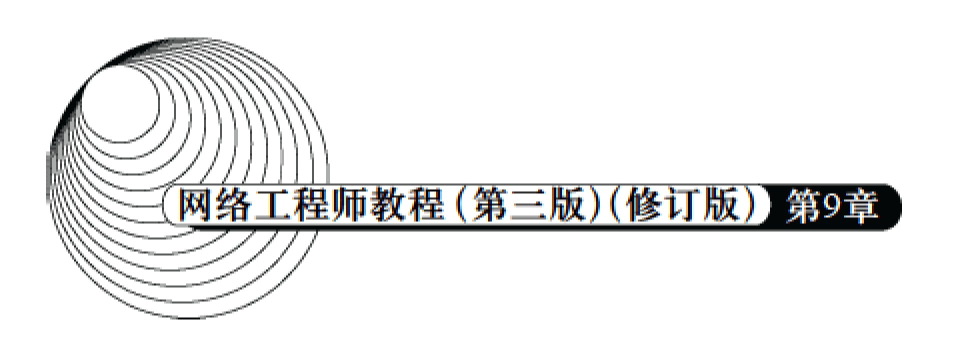
对以上参数解释如下。

- -a *RemoteName*

显示远程计算机的 NetBIOS 名字表，用标识符 RemoteName 指示远程计算机的名字。

- -A *IPAddress*

显示远程计算机的 NetBIOS 名字表，用标识符 IPAddress 指示远程计算机的 IP 地址。



- -c

显示 NetBIOS 名字缓存的内容。

- -n

显示本地计算机的 NetBIOS 名字表。

- -r

显示 NetBIOS 名字解析的统计数据。在配置了 WINS 的 Windows XP 计算机上，这个参数返回通过广播解析的名字，以及通过 WINS 服务器解析的名字。

- -R

清除 NetBIOS 名字缓存，并从 Lmhosts 文件装载带有标签#PRE 的预加载项目。

- -RR

释放并刷新本地计算机在 WINS 服务器中注册的名字。

- -s

显示 NetBIOS 客户端与服务器的会话，并把目标 IP 地址转换为名字。

- -S

显示 NetBIOS 客户端与服务器的会话，用 IP 地址表示远程计算机。

- *Interval*

多次显示统计数据，显示的间隔时间由标识符 Interval（秒）表示，直至输入 Ctrl+C 停止显示。如果这个参数缺失，只显示一次。

Nbtstat 命令行参数是大小写敏感的，所以-A，-R，-RR 和-S 等必须大写。

表 9-8 表示 nbtstat 命令显示的列表栏目的含义。表 9-9 说明了 NetBIOS 连接的状态。

表 9-8　nbtstat 列表栏目的含义

栏　　目	解　　释
Input	接收的字节数
Output	发送的字节数
In/Out	连接是入径（inbound）或出径（outbound）
Life	名字缓存表项的剩余生命期
Local Name	NetBIOS 连接的本地名字
Remote Host	远程计算机的名字或地址
Type	名字的类型，可以是唯一名字（unique）或组名字（group）
Status	已注册（Registered），冲突（Conflict）
State	NetBIOS 连接的状态

表 9-9　NetBIOS 连接的状态

状　　态	解　　释
Connected	会话已经建立
Associated	连接端点已经产生，并分配了一个 IP 地址
Listening	端点正在等待入径连接
Idle	端点已经打开，但不能解释连接
Connecting	会话处于建立阶段，正在解析目标的名字——地址映射
Accepting	正在解释一个入径会话，连接很快就要建立
Reconnecting	一个会话正在重新连接
Outbound	一个会话处于正在建立连接阶段，TCP 连接已经生成
Inbound	一个入径会话处于建立连接阶段
Disconnecting	会话正在断开阶段
Disconnected	本地计算机发出了释放连接请求，正在等待远端系统的应答

举例如下。

（1）要显示远端计算机 CORP07 的 NetBIOS 名字表，则输入：

```
nbtstat -a CORP07
```

（2）要显示地址为 10.0.0.99 的远端计算机的 NetBIOS 名字表，则输入：

```
nbtstat -A 10.0.0.99
```

（3）要显示本地计算机的 NetBIOS 名字表，则输入：

```
nbtstat -n
```

（4）要显示本地计算机 NetBIOS 名字缓存的内容，则输入：

```
nbtstat -c
```

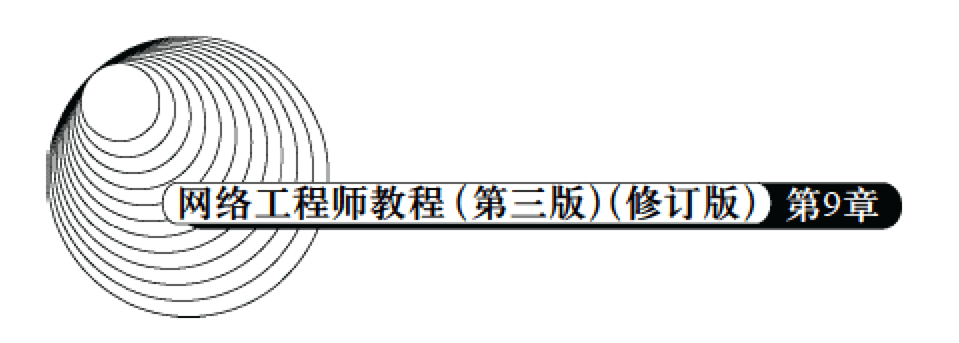
（5）要清除 NetBIOS 名字缓存，并从本地 Lmhosts 文件重装预加载项目，则输入：

```
nbtstat -R
```

（6）要释放本地计算机在 WINS 服务器中注册的 NetBIOS 名字并重新注册，则输入：

```
nbtstat -RR
```

（7）要显示 NetBIOS 会话统计数据，每 5 s 显示一次，则输入：



nbtstat -S 5

9.7.8 Route

这个命令的功能是显示和修改本地的 IP 路由表，如果不带参数，则给出帮助信息。Route 命令的语法如下：

route [-f] [-p] [*Command* [*Destination*] [mask *Netmask*] [*Gateway*] [metric *Metric*]] [if *Interface*]]

对以上参数解释如下。

- -f

删除路由表中的网络路由（子网掩码不是 255.255.255.255）、本地环路路由（目标地址为 127.0.0.0，子网掩码为 255.0.0.0）和组播路由（目标地址为 224.0.0.0，子网掩码为 240.0.0.0）。如果与其他命令（例如 add、change 或 delete）联合使用，在运行这个命令前先清除路由表。

- -p

与 add 命令联合使用时，一条路由被添加到注册表中，当 TCP/IP 协议启动时，用于初始化路由表。在默认情况下，系统重新启动时不保留添加的路由。与 print 命令联合使用时，则显示持久路由列表。对于其他命令，这个参数被忽略。持久路由保存在注册表中的 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters \PersistentRoutes 位置。

- *Command*

表示要运行的命令，可用的命令如表 9-10 所示。

表 9-10　可用的命令

命　　令	用　　途	命　　令	用　　途
add	添加路由	delete	删除路由
change	修改已有的路由	print	打印路由

- *Destination*

说明目标地址，可以是网络地址（IP 地址中对应主机的位都是 0）、主机地址或默认路由（0.0.0.0）。

- mask *Netmask*

说明了目标地址对应的子网掩码。网络地址的子网掩码依据网络的大小而变化，主机地址的子网掩码为 255.255.255.255，默认路由的子网掩码为 0.0.0.0。如果忽略了这个参数，默认的子网掩码为 255.255.255.255。由于在路由寻址中具有关键作用，所以目标地址不能特异于对应的子网掩码。换言之，如果子网掩码的某位是 0，则目标地址的对应位不能为 1。

- Gateway

说明下一跃点的 IP 地址。对于本地连接的子网，网关地址是本地子网中分配给接口的 IP 地址。对于远程路由，网关地址是相邻路由器中直接连接的 IP 地址。

- metric *Metric*

说明路由度量值（1～9999）。通常选择度量值最小的路由。度量值可以根据跃点数、链路速率通路可靠性、通路的吞吐率以及管理属性等参数确定。

- if *Interface*

说明接口的索引。使用 route print 命令可以显示接口索引列表。接口索引可以使用十进制数或十六进制数表示。如果忽略 if 参数，接口索引根据网关地址确定。

路由表中可能出现很大的度量值，这是 TCP/IP 协议根据 LAN 接口配置的 IP 地址、子网掩码和默认网关等参数自动计算的度量值。自动计算接口度量值是默认的，就是根据接口的速率调整路由度量，所以最快的接口生成了最低的度量值。要消除大的度量值，就要应用“高级 TCP/IP 设置”来废除“自动跃点计数”选项，如图 9-46 所示。

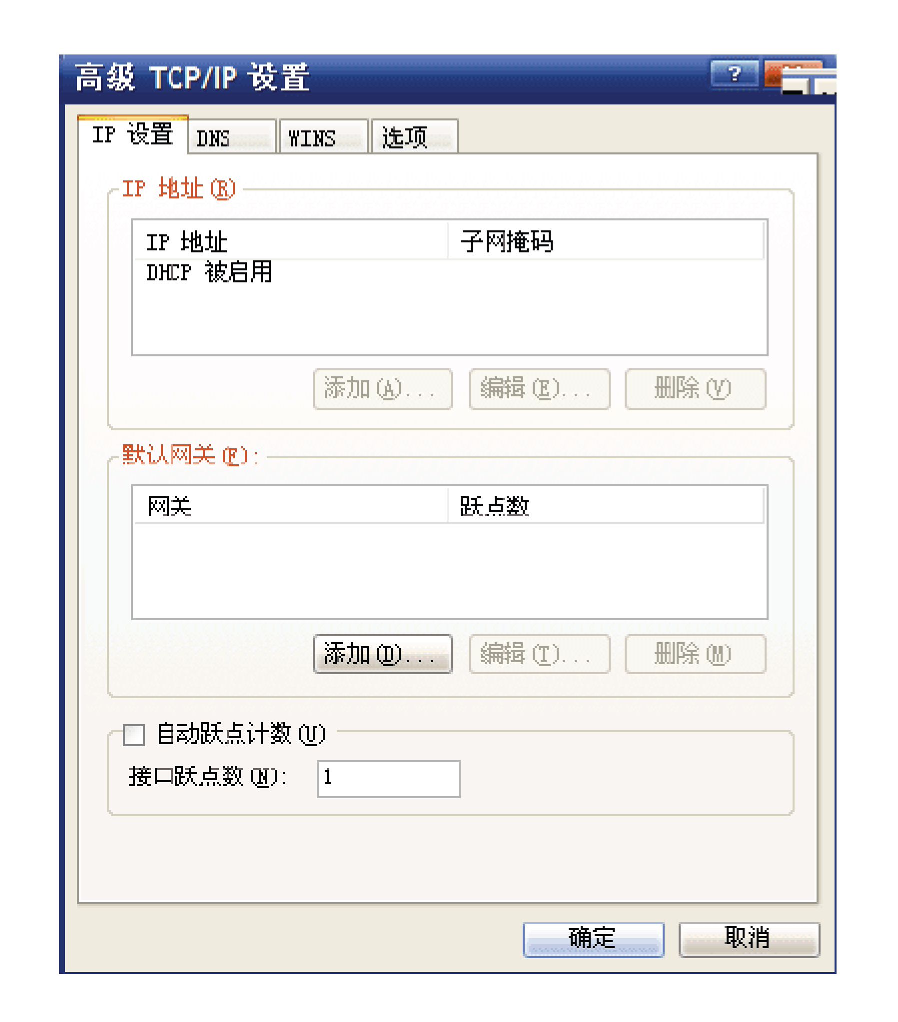


图 9-46　高级 TCP/IP 设置

可以用名字表示路由目标，如在%Systemroot%\System32\Dtivers\Etc\hosts 或 Lmhosts 文件中存在相应表项的话。也可以用名字表示网关，只要这个名字可以通过标准方法解析为 IP 地址。

在使用命令 print 或 delete 时可以忽略参数 Gateway，使用通配符来代替目标和网关。目标可以用一个星号“*”来代替。如果目标的值中包含星号“*”或问号“？”，也被看作是通配符，用于匹配被打印或被删除的目标路由。事实上，星号可以匹配任何字符串，问号则用于匹配任何单个字符。例如，10.*.1、192.168.*和*224*都是合法的通配符。

如果使用了目标地址与子网掩码的无效组合，则会显示“Route: bad gateway address netmask”的错误信息。当目标地址中的一个或多个位被设置为“1”，而子网掩码的对应位却被设置为“0”时，就会出现这种错误。为了检查这种错误，可以把目标地址和子网掩码都用二进制表示。子网掩码的二进制表示中，开头有一串“1”，代表网络地址部分，后跟一串“0”，代表主机地址部分。这样就可以确定，是否目标地址中属于主机的位被设置成了“1”。

-p 参数只能在 Windows NT 4.0、Windows 2000/2003、Windows Millennium Edition 和 Windows XP 中使用，Windows 9x 不支持这个参数。

举例如下。

（1）要显示整个路由器的内容，则输入：

route print

（2）要显示路由表中以 10.开头的表项，则输入：

route print 10.*

（3）对网关地址 192.168.12.1 要添加一条默认路由，则输入：

route add 0.0.0.0 mask 0.0.0.0 192.168.12.1

（4）要添加一条到达目标 10.41.0.0（子网掩码为 255.255.0.0）的路由，下一跃点地址为 10.27.0.1，则输入：

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1

（5）要添加一条到达目标 10.41.0.0（子网掩码为 255.255.0.0）的持久路由，下一跃点地址为 10.27.0.1，则输入：

route -p add 10.41.0.0 mask 255.255.0.0 10.27.0.1

（6）要添加一条到达目标 10.41.0.0 255.255.0.0 的路由，下一跃点地址为 10.27.0.1，度量值为 7，则输入：

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 metric 7

（7）要添加一条到达目标 10.41.0.0 255.255.0.0 的路由，下一跃点地址为 10.27.0.1，接口索引为 0x3，则输入：

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 if 0x3

（8）要删除到达目标 10.41.0.0 255.255.0.0 的路由，则输入：

route delete 10.41.0.0 mask 255.255.0.0

（9）要删除路由表中所有以 10.开头的表项，则输入：

route delete 10.*

（10）要把目标 10.41.0.0 255.255.0.0 的下一跃点地址由 10.27.0.1 改为 10.27.0.25，则输入：

route change 10.41.0.0 mask 255.255.0.0 10.27.0.25

9.7.9 Netsh

Netsh 是一个命令行脚本实用程序，可用于修改计算机的网络配置。利用 Netsh 也可以建立批文件来运行一组命令，或者把当前的配置脚本用文本文件保存起来，以后可用来配置其他的服务器。

1. Netsh 上下文

Netsh 利用动态链接库（DLL）与操作系统的其他组件交互作用。Netsh 助手（helper ）是一种动态链接库文件，提供了称为上下文（context）的扩展特性，这是一组可作用于某种网络组件的命令。Netsh 上下文扩大了它的作用，可以对多种服务、实用程序或协议提供配置和监控功能。例如，Dhcpmon.dll 就是一种 Netsh 助手文件，它提供了一组配置和管理 DHCP 服务器的命令。

运行 Netsh 命令要从 Cmd.exe 提示符开始，然后转到指定的上下文。可使用的上下文取决于已经安装的网络组件。例如，在 Netsh 命令提示符（netsh>）下输入 dhcp，就会转到 DHCP 上下文。但是如果没有安装 DHCP 服务，则会出现下面的信息：

```
The following command was not found: dhcp.
```

2. 使用多个上下文

从一个上下文可以转到另一个上下文，后者叫做子上下文。例如，在路由上下文中可以转到 IP 或 IPX 上下文。

为了显示在某个上下文中可使用的子上下文和命令列表，可以在 Netsh 提示符下输入上下文的名字，后跟“？”或 help。例如，为了显示在路由上下文中可使用的子上下文和命令，在 netsh 提示符下输入：

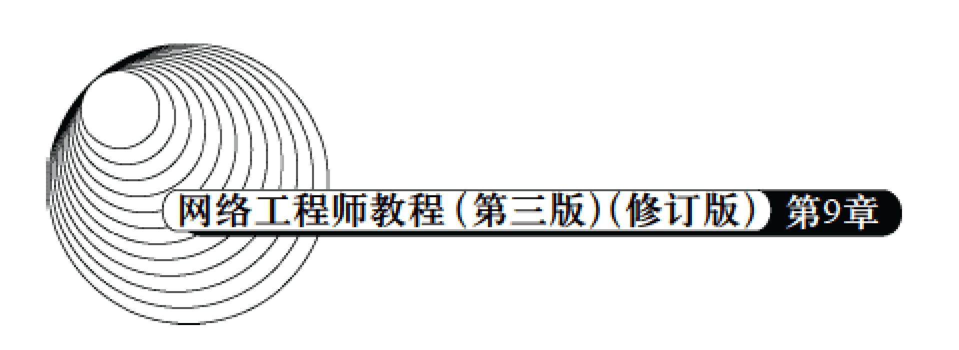
```
netsh>routing ?
```

或者

```
netsh>routing help
```

为了不改变当前上下文而完成另外一个上下文中的任务，可以在 Netsh 提示符下输入命令的上下文路径。例如，要在 IGMP 上下文中添加“本地连接”接口而不改变到 IGMP 上下文，则输入：

```
netsh>routing ip igmp add interface "Local Area Connection" startupqueryinterval=21
```



3．在 Cmd.exe 命令提示符下运行 Netsh 命令

为了在远程 Windows Server 2003 中运行 Netsh 命令，首先要通过“远程桌面连接”连接到正在运行终端服务器的 Windows Server 2003 系统中。在 Cmd.exe 命令提示符下输入 netsh，就进入了 netsh> 提示符。Netsh 的语法如下：

netsh [-a *AliasFile*] [-c *Context*] [-r *RemoteComputer*] [{*NetshCommand*|-f *ScriptFile*}]

对以上参数解释如下：

- -a *AliasFile*

运行 AliasFile 文件后返回 netsh 提示符。

- -c *Context*

转到指定的 netsh 上下文，可用的上下文如表 9-11 所示。

表 9-11 netsh 上下文

上　下　文	解　　释
AAAA	配置认证、授权、计费和审计（Authentication, Authorization, Accounting, and Auditing, AAAA）数据库，该数据库是 Internet 认证服务器和路由及远程访问服务器要使用的
DHCP	管理 DHCP 服务器
Diag	操作系统和网络服务的管理及故障诊断
Interface	配置 TCP/IP 协议，显示配置和统计信息
RAS	管理远程访问服务器
Routing	管理路由服务器
WINS	管理 WINS 服务器

- -r *RemoteComputer*

配置远程计算机。

- *NetshCommand*

说明要使用的 netsh 命令。

- -f *ScriptFile*

运行脚本后转出 netsh.exe。

关于-r 参数的使用值得注意。如果在-r 参数中使用了另外的命令，则 netsh 在远程计算机上执行这个命令，然后返回到 cmd.exe 命令提示符下。如果使用-r 参数而没有使用其他命令，则 netsh 保持在远程模式。这个过程类似于在 netsh 命令提示符下执行 set machine 命令。在使用-r 参数时，只是在当前的 netsh 实例中配置目标机器。在转出并重新进入 netsh 后，目标机器

又变成了本地计算机。远程计算机的名字可以是存储在 WINS 服务器上的名字、UNC（Universal Naming Convention）名字、可以被 DNS 服务器解析的 Internet 名字或者 IP 地址。

4. 在 Netsh.exe 提示符下运行 Netsh 命令

在 netsh>提示符下可以使用下面一些命令。

- .. ：转移到上一层上下文。
- abort：放弃在脱机模式下所做的修改。
- add helper *DLLName*：在 netsh 中安装 netsh 助手文件 DLLName。
- alias [*AliasName*]：显示指定的别名。

alias [*AliasName*][*string1* [*string2*…]]：设置 AliasName 的别名为指定的字符串。

可以使用别名命令行替换 netsh 命令，或者将其他平台中更熟悉的命令映射到适当的 netsh 命令。下面是使用 alias 的例子，这个脚本设置了两个别名 Shaddr 和 Shp，并进入 netsh interface ip 上下文：

```
alias shaddr show interface ip addr
alias shp show helpers
interface ip
```

如果在 Netsh 命令提示符下输入 shaddr，则被解释为命令 show interface ip addr；如果在 Netsh 命令提示符下输入 shp，则被解释为命令 show helpers。

- bye：转出 Netsh。
- commit：向路由器提交在脱机模式下所做的改变。
- delete helper *DLLName*：删除 netsh 助手文件 DLLName。
- dump [*FileName*]：生成一个包含当前配置的脚本。如果要把脚本保存在文件中，则使用参数 FileName。如果不带参数，则显示当前配置脚本。
- exec *ScriptFile*：装载并运行脚本文件 ScriptFile。脚本文件运行在一个或多个计算机上。
- exit：从 Netsh 转出。
- help：显示帮助信息，可以用/?或?或 h 代替。
- offline：设置为脱机模式。
- online：设置为联机模式。

在脱机模式下做出的配置可以保存起来，通过运行 commit 命令或联机命令在路由器上执行。从脱机模式转到联机模式时，在脱机模式下做出的改变会反映在当前正在运行的配置中，而在联机模式下做出的改变会立即反映在当前正在运行的配置中。

- popd：从堆栈中恢复上下文。
- pushd：把当前的上下文保存在堆栈中。

popd 与 pushd 配合使用，可以改变到新的上下文，运行新的命令，然后恢复前面的上下文。下面是使用这两个命令的例子。这个脚本首先从根脚本转到 interface ip 上下文，添加一个静态路由，然后返回根上下文。

```
netsh>
pushd
netsh>
interface ip
netsh interface ip>
set address local static 10.0.0.9 255.0.0.0 10.0.0.1 1
netsh interface ip>
popd
netsh>
```

- quit：转出 Netsh。
- set file {open *FileName*|append *FileName*| *close*}：拷贝命令提示符窗口的输出到指定的文件。其中的参数如下。
  - ◆ open *FileName*：打开文件 FileName，并发送命令提示符窗口的输出到这个文件。
  - ◆ append *FileName*：附加命令提示符窗口的输出到指定的文件 FileName。
  - ◆ Close：停止发送输出并关闭文件。

如果指定的文件不存在，则 netsh 生成一个新文件；如果指定的文件存在，则 netsh 重写文件中已有的数据。下面的命令生成一个叫做 session.log 的记录文件，并拷贝 netsh 的输入和输出到这个文件：

```
set file open c:\session.log
```

- set machine [[*ComputerName*=]*string*]：指定当前要完成配置任务的计算机，其中的字符串 string 是远程计算机的名字。如果不带参数，则指本地计算机。

在一个脚本中，可以在多个计算机上执行命令。在一个脚本中，首先利用 set machine 命令说明一个计算机 ComputerA，在这个计算机上运行随后的命令。然后再利用 set machine 命令指定另外一个计算机 ComputerB，再在这个计算机上运行命令。

- set mode {online|offline}：设置为联机或脱机模式。
- show {alias|helper|mode}：显示别名、助手或当前的模式。
- unalias *AliasName*：删除指定的别名。

9.7.10 Nslookup

Nslookup 命令用于显示 DNS 查询信息，诊断和排除 DNS 故障。使用这个工具必须熟悉 DNS 服务器的工作原理（参见本书第 7 章）。Nslookup 有交互式和非交互式两种工作方式。Nslookup 的语法如下：

- nslookup [-option ...]　　#使用默认服务器，进入交互方式
- nslookup [-option ...] −server　　#使用指定服务器 server，进入交互方式
- nslookup [-option ...] host　　#使用默认服务器，查询主机信息
- nslookup [-option ...] host server　　#使用指定服务器 Server，查询主机信息
- ? | /? | /help　　#显示帮助信息

1．非交互式工作

所谓非交互式工作，就是只使用一次 Nslookup 命令后又返回到 Cmd.exe 提示符下。如果只查询一项信息，可以进入这种工作方式。Nslookup 命令后面可以跟随一个或多个命令行选项（option），用于设置查询参数。每个命令行选项由一个连字符“-”后跟选项的名字，有时还要加一个等号“=”和一个数值。

在非交互方式中，第一个参数是要查询的计算机（host）的名字或 IP 地址，第二个参数是 DNS 服务器（server）的名字或 IP 地址，整个命令行的长度必须小于 256 个字符。如果忽略了第二个参数，则使用默认的 DNS 服务器。如果指定的 host 是 IP 地址，则返回计算机的名字；如果指定的 host 是名字，并且没有尾随的句点，则默认的 DNS 域名被附加在后面（设置了 defname），查询结果给出目标计算机的 IP 地址。如果要查找不在当前 DNS 域中的计算机，在其名字后面要添加一个句点“.”（称为尾随点）。下面举例说明非交互方式的用法。

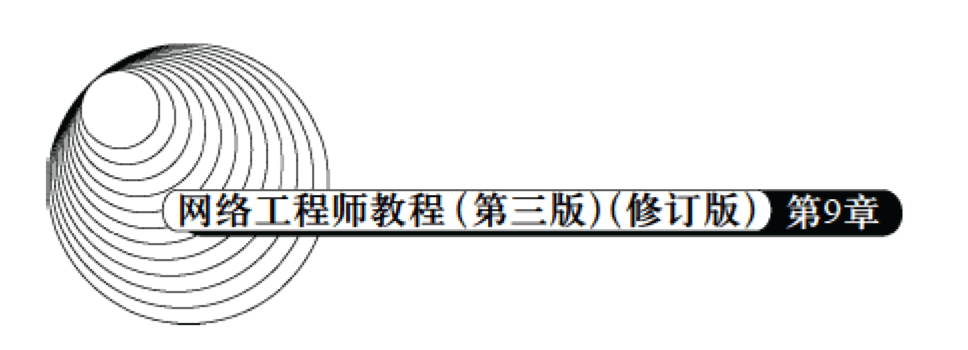
（1）应用默认的 DNS 服务器根据域名查找 IP 地址。

```
C:\>nslookup ns1.isi.edu
Server: ns1.domain.com
Address: 202.30.19.1

Non-authoritative answer:        #给出应答的服务器不是该域的权威服务器
Name: ns1.isi.edu
Address: 128.9.0.107             #查出的 IP 地址
```

（2）应用默认的 DNS 服务器根据 IP 地址查找域名。

```
C:\>nslookup 128.9.0.107
```




```
Server: ns1.domain.com
Address: 202.30.19.1

Name: ns1.isi.edu                #查出的 IP 地址
Address: 128.9.0.107
```

（3）Nslookup 命令后面可以跟随一个或多个命令行选项（option）。例如，要把默认的查询类型改为主机信息，把超时间隔改为 5 s，查询的域名为 ns1.isi.edu，则使用下面的命令：

```
C:\>nslookup -type=hinfo -timeout=5 ns1.isi.edu
Server: ns1.domain.com
Address: 202.30.19.1

isi.edu                                       #给出了 SOA 记录
     primary name server = isi.edu            #主服务器
     responsible mail addr = action.isi.edu   #邮件服务器
     serial = 2009010800                      #查询请求的序列号
     refresh     = 7200 <2 hours>             #刷新时间间隔
     retry  = 1800 <30 mins>                  #重试时间间隔
     expire      = 604800 <7 days>            #辅助服务器更新有效期
     default TTL = 86400 <1 days>             #资源记录在 DNS 缓存中的有效期
C:\>
```

2．交互式工作

如果需要查找多项数据，可以使用 Nslookup 的交互工作方式。在 Cmd.exe 提示符下输入 nslookup 后按 Enter 键，就进入了交互工作方式，命令提示符变成“>”。

在命令提示符“>”下输入 help 或?，会显示可用的命令列表（如图 9-47 所示）；如果输入 exit，则返回 Cmd.exe 提示符。

在交互方式下，可以用 set 命令设置选项，满足指定的查询需要。下面举出几个常用子命令的应用实例。

（1）>set all：列出当前设置的默认选项。

```
>set all
Server: ns1.domain.com
Address: 202.30.19.1

Set options:
```

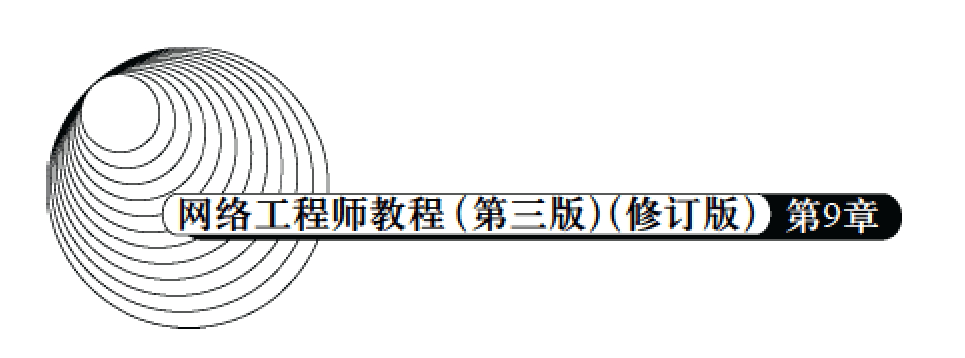
```
nodebug                              #不打印排错信息
defname                              #对每一个查询附加本地域名
search                               #使用域名搜索列表
...........................（省略）.................................
MSxfr                                #使用 MS 快速区域传输
IXFRversion=1                        #当前的 IXFR（渐增式区域传输）版本号
srchlist=                            #查询搜索列表
```

```
Commands: (identifiers are shown in uppercase, [] means optional)
NAME - print info about the host/domain NAME using default server
NAME1 NAME2 - as above, but use NAME2 as server
help or ? - print info on common commands
set OPTION - set an option
    all - print options, current server and host
    [no]debug - print debugging information
    [no]d2 - print exhaustive debugging information
    [no]defname - append domain name to each query
    [no]recurse - ask for recursive answer to query
    [no]search - use domain search list
    [no]vc - always use a virtual circuit
    domain=NAME - set default domain name to NAME
    srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1, N2, etc.
    root=NAME - set root server to NAME
    retry=X - set number of retries to X
    timeout=X - set initial time-out interval to X seconds
    type=X - set query type (for example, A, ANY, CNAME, MX, NS, PTR, SOA, SRV)
    querytype=X - same as type
    class=X - set query class (for example, IN (Internet), ANY)
    [no]msxfr - use MS fast zone transfer
    ixfrver=X - current version to use in IXFR transfer request
server NAME - set default server to NAME, using current default server
lserver NAME - set default server to NAME, using initial server
finger [USER] - finger the optional NAME at the current default host
root - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
    -a - list canonical names and aliases
    -d - list all records
    -t TYPE - list records of the given type (for example, A, CNAME, MX, NS, PTR, and so on)
view FILE - sort an 'ls' output file and view it with pg
exit - exit the program
```

图 9-47　nslookup 子命令

（2）set type=mx：这个命令查询本地域的邮件交换器信息。

```
C:\> nslookup
```




```
Default Server: ns1.domain.com
Address: 202.30.19.1
> set type=mx
> 163.com.cn
Server: ns1.domain.com
Address: 202.30.19.1

Non-authoritative answer:
163.com.cn        MX preference = 10, mail exchanger =mx1.163.com.cn
163.com.cn        MX preference = 20, mail exchanger =mx2.163.com.cn
mx1.163.com.cn internet address = 61.145.126.68
mx2.163.com.cn internet address = 61.145.126.30
>
```

（3）server NAME：由当前默认服务器切换到指定的名字服务器 NAME。类似的命令 lserver 是由本地服务器切换到指定的名字服务器。

```
C:\> nslookup
Default Server: ns1.domain.com
Address: 202.30.19.1
> server 202.30.19.2
Default Server: ns2.domain.com
Address: 202.30.19.2
```

（4）ls：这个命令用于区域传输，罗列出本地区域中的所有主机信息。ls 命令的语法如下：

```
ls [- a |-d | -t type] domain [> filename]
```

不带参数使用 ls 命令将显示指定域（domain）中所有主机的 IP 地址。-a 参数返回正式名称和别名，-d 参数返回所有数据资源记录，而-t 参数将列出指定类型（type）的资源记录。任选的 filename 是存储显示信息的文件。如图 9-48 所示。

如果安全设置禁止区域传输，将返回下面的错误信息：

```
*** Can't list domain example.com ： Server failed
```

（5）set type：该命令的作用是设置查询的资源记录类型。DNS 服务器中主要的资源记录有 A（域名到 IP 地址的映射）、PTR（IP 地址到域名的映射）、MX（邮件服务器及其优先级）、CNAM（别名）和 NS（区域的授权服务器）等类型。通过 A 记录可以由域名查地址，也可以由地址查域名。在图 9-49 中，用 set all 命令显示默认设置，可以看出 type=A+AAAA，这时可以进行正向查询，也可以进行反向查询，如图 9-50 所示。

```
> ls xidian.edu.cn
[ns1.xidian.edu.cn]
 xidian.edu.cn.                 NS      server = ns1.xidian.edu.cn
 xidian.edu.cn.                 NS      server = ns2.xidian.edu.cn
 408net                         A       202.117.118.25
 acc                            A       202.117.121.5
 ai                             A       202.117.121.146
 antanna                        A       219.245.110.146
 apweb2k                        A       202.117.116.19
 bbs                            A       202.117.112.11
 cce                            A       210.27.3.95
 cese                           A       219.245.118.199
 cnc                            A       210.27.5.123
 cnis                           A       202.117.112.16
 www.cnis                       A       202.117.112.16
 con                            A       202.117.112.6
 cpi                            A       219.245.78.155
 cs                             A       202.117.112.23
 csti                           A       202.117.114.31
 cwc                            A       210.27.1.33
 cxjh                           A       202.117.112.27
 Dec586                         A       202.117.112.15
 dingzhg                        A       202.117.117.8
 djzx                           A       202.117.121.87
 dp                             A       210.27.12.227
 dtg                            A       202.117.114.35
 dttrdc                         A       219.245.79.48
 ecard                          A       202.117.112.199
 ecm                            A       202.117.116.79
 ecr                            A       202.117.115.9
 ee                             A       210.27.6.158
```

图 9-48　ls 命令的输出

```
> server 61.134.1.4          #设置默认服务器
默认服务器:  [61.134.1.4]
Address:  61.134.1.4

> set all
默认服务器:  [61.134.1.4]
Address:  61.134.1.4

设置选项:
  nodebug
  defname
  search
  recurse
  nod2
  novc
  noignoretc
  port=53
  type=A+AAAA              #查询 A 记录和 AAAA记录
  class=IN                  可以给出 IPv4 和 IPv6 地址
  timeout=2
  retry=1
  root=A.ROOT-SERVERS.NET.
  domain=
  MSxfr
  IXFRversion=1
  srchlist=
```

图 9-49　set all 显示默认设置

```
> www.tsinghua.edu.cn
服务器:  [61.134.1.4]
Address:  61.134.1.4

非权威应答:
名称:    www.d.tsinghua.edu.cn
Addresses:  2001:da8:200:200::4:100
          211.151.91.165
Aliases:  www.tsinghua.edu.cn

> 211.151.91.165
服务器:  [61.134.1.4]
Address:  61.134.1.4

名称:    165.tsinghua.edu.cn
Address:  211.151.91.165
```

图 9-50　查询 A 记录和 AAAA 记录

当查询 PTR 记录时，可以由地址查到域名，但是没有从域名查到地址，而是给出了 SOA 记录，如图 9-51 所示。

```
> set type=ptr                                                  # 查询PTR记录
> 211.151.91.165                                                # 由地址查域名
服务器:  [61.134.1.4]
Address:  61.134.1.4

非权威应答:
165.91.151.211.in-addr.arpa     name = 165.tsinghua.edu.cn      # 查询成功，得到域名
> www.tsinghua.edu.cn                                           # 由域名查地址
服务器:  [61.134.1.4]
Address:  61.134.1.4

DNS request timed out.
    timeout was 2 seconds.
非权威应答:
www.tsinghua.edu.cn     canonical name = www.d.tsinghua.edu.cn

d.tsinghua.edu.cn
        primary name server = dns.d.tsinghua.edu.cn             # 没有查出地址
        responsible mail addr = szhu.dns.edu.cn                   但给出了SOA记录
        serial  = 2007042815
        refresh = 3600 (1 hour)
        retry   = 1800 (30 mins)
        expire  = 604800 (7 days)
        default TTL = 86400 (1 day)
```

图 9-51　查询 PTR 记录

重新查询 A 记录，可以进行双向查询，如图 9-52 所示。

（6）set type=any：对查询的域名显示各种可用的信息资源记录（A、CNAME、MX、NS、PTR、SOA 和 SRV 等），如图 9-53 所示。

```
> set type=a                                #查询 A 记录
> www.tsinghua.edu.cn                       #由域名查地址
服务器:  [61.134.1.4]
Address:  61.134.1.4

非权威应答:
名称:    www.d.tsinghua.edu.cn
Address:  211.151.91.165                    #查出地址，并出给别名
Aliases:  www.tsinghua.edu.cn

> 211.151.91.165                            #由地址查域名
服务器:  [61.134.1.4]
Address:  61.134.1.4

名称:    165.tsinghua.edu.cn                #查询成功，得到域名
Address:  211.151.91.165

> _
```

图 9-52　查询 A 记录

```
> set type=any
> baidu.com
服务器:  [218.30.19.40]
Address:  218.30.19.40

非权威应答:
baidu.com       internet address = 202.108.23.59
baidu.com       internet address = 220.181.5.97
baidu.com       nameserver = dns.baidu.com
baidu.com       nameserver = ns2.baidu.com
baidu.com       nameserver = ns3.baidu.com
baidu.com       nameserver = ns4.baidu.com
baidu.com       MX preference = 10, mail exchanger = mx1.baidu.com
>
```

图 9-53　各种信息资源记录

（7）set degug：这个命令与 set d2 的作用类似，都是显示查询过程的详细信息，set d2 显示的信息更多，有查询请求报文的内容和应答报文的内容。图 9-54 是利用 set d2 显示的查询过程。这些信息可用于对 DNS 服务器进行排错。

9.7.11　Net

Windows 中的网络服务都使用以 net 开头的命令。在 Cmd.exe 提示符下输入 net /?，则显示 net 命令的列表如下：

NET [ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP |
　　HELPMSG | LOCALGROUP | NAME | PAUSE | PRINT | SEND | SESSION |
　　SHARE | START | STATISTICS | STOP | TIME | USE | USER | VIEW]

如果要查看某个 net 命令的使用方法，则输入 net help “命令名”。例如为显示 accounts 命令的用法，输入 c:\ >net help accounts，结果如图 9-55 所示。

```
> set d2
> 163.com.cn
服务器:  UnKnown
Address:  218.30.19.40

------------
SendRequest(), len 28
    HEADER:
        opcode = QUERY, id = 2, rcode = NOERROR
        header flags:  query, want recursion
        questions = 1,  answers = 0,  authority records = 0,  additional = 0

    QUESTIONS:
        163.com.cn, type = A, class = IN

------------
------------
Got answer (44 bytes):
    HEADER:
        opcode = QUERY, id = 2, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        163.com.cn, type = A, class = IN
    ANSWERS:
    ->  163.com.cn
        type = A, class = IN, dlen = 4
        internet address = 219.137.167.157
        ttl = 86400 (1 day)
------------
非权威应答:
------------
SendRequest(), len 28
    HEADER:
        opcode = QUERY, id = 3, rcode = NOERROR
        header flags:  query, want recursion
        questions = 1,  answers = 0,  authority records = 0,  additional = 0

    QUESTIONS:
        163.com.cn, type = AAAA, class = IN

------------
------------
Got answer (28 bytes):
    HEADER:
        opcode = QUERY, id = 3, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 0,  authority records = 0,  additional = 0

    QUESTIONS:
        163.com.cn, type = AAAA, class = IN

------------
名称:    163.com.cn
Address:  219.137.167.157

>
```

图 9-54　显示查询过程

下面举出几个常用的 net 命令的例子。

- c:\>net user：显示所有用户的列表。
- c:\>net share：显示共享资源。
- c:\>net start：显示已启动的服务列表。
- c:\>net start telnet：启动 telnet 服务。
- c:\>net stop telnet：停止 telnet 服务。

```
C:\Documents and Settings\Administrator>net help accounts
此命令的语法是:


NET ACCOUNTS
[/FORCELOGOFF:{minutes | NO}] [/MINPWLEN:length]
             [/MAXPWAGE:{days | UNLIMITED}] [/MINPWAGE:days]
             [/UNIQUEPW:number] [/DOMAIN]


NET ACCOUNTS 命令用于更新用户的帐户数据库，并为所有帐户修改密码
和登录需求。当在不加选项的情况下使用这个命令时，NET ACCOUNTS 会
显示密码，登录限制，以及域信息的当前设置。

为了使用带有选项的 NET ACCOUNTS 命令，需要如下两个条件:

*  仅当用户的帐户已经设立时(使用“用户管理器”或 NET USER 命令)，密码和
   登录需求才会起作用。

*  所有验证登录的域服务器必须运行 NET Logon 服务。当 Windows 启动时，
   Net Logon 会自动启动。

/FORCELOGOFF:{minutes | NO}  设置用户被强迫退出系统之前所拥有的分钟数。这
                             种情况会在帐户过期或有效的登录时间过期时出现。
                             默认值是 NO，表示禁止强迫退出系统。
/MINPWLEN:length             设置密码的最少字符数。字符数的范围是0-14个字
                             符；默认值是 6 个字符。
/MAXPWAGE:{days | UNLIMITED} 设置密码有效的最大天数。用 UNLIMITED 指定没有
                             限制。/MAXPWAGE 选项不能小于 /MINPWAGE。
                             其范围是 1-999。默认值为保持此值不变。
/MINPWAGE:days               设置用户不能改变密码的最小天数。0 表示没有该
                             限制。其范围是 0-999；默认值是 0 天。/MINPWAGE
                             选项不能大于 /MAXPWAGE 选项。
/UNIQUEPW:number             要求用户的密码在指定的密码更改次数内必须保持唯一。

                             其最大值是 24。
/DOMAIN                      在当前域的主域控制器上执行操作。否则在本地计算
                             机上执行操作。

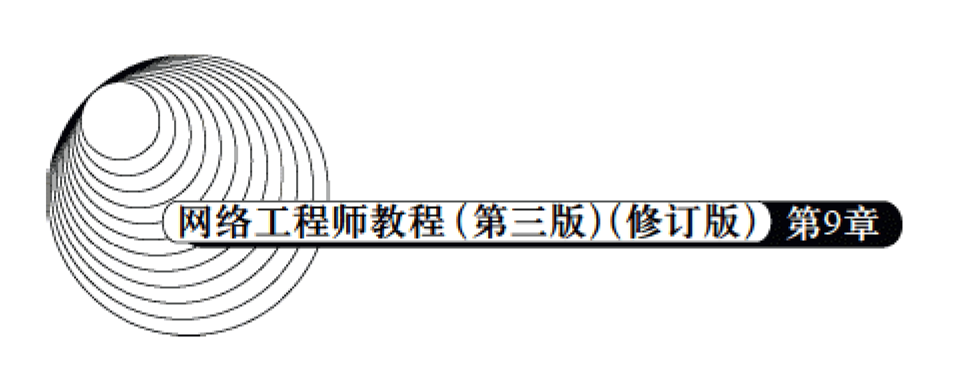

NET HELP command | MORE 逐屏显示帮助。
```

图 9-55　net 帮助命令

- c:\>net use：显示已建立的网络连接。
- c:\>net view：显示计算机上的共享资源列表。
- c:\>net send：192.16.810.1 “时间到了，请关机”向地址为 192.168.10.1 的计算机发送消息。

9.8　网络监视和管理工具

用于采集网络数据流并提供数据分析能力的工具称为网络监视器。监视网络的目的是对数据流进行分析，发现网络通信中的问题。网络监视器能提供利用率和数据流量方面的统计数据，还能从网络通信流中捕获数据帧，并筛选、解释、分析这些数据帧的内容，判断其来源和去向。目前大多数网络都是基于以太网构建的，广播通信方式决定了在一台计算机上可以采集到子网内的全部通信流，因此网络监视器的有效范围遍及路由器以内的全部通信主机。



目前最常用的网络监视工具有 Sniffer、NetXray 和 Ethereal 等，其中 Sniffer 的功能最强，使用最为普遍。下面介绍 Sniffer 的功能和使用方法。

9.8.1 网络监听原理

由于以太网采用广播通信方式，所以在网络中传送的分组可以出现在同一冲突域中的所有端口上。在常规状态下，网卡控制程序只接收发送给自己的数据包和广播包，对目标地址不是自己的数据包则丢弃。如果把网卡配置成混杂模式（Promiscuous Mode），它就能接收所有分组，无论是否是发送给自己的。

采用混杂模式的程序可以把网络连接上传输的所有分组都显示在屏幕上。有些协议（例如 FTP 和 Telnet）在传输数据和口令字时不进行加密，采用混杂模式的网络扫描器就可以解读和提取有用的信息，这给网络黑客造成了可乘之机。利用网络监听技术，既可以进行网络监控，解决网络管理中的问题，也可以进行网络窃听，实现网络入侵的目的。

当一个主机采用混杂模式进行网络监听时，它是可以被检查出来的。这里主要有两种方法：一种是根据时延来判断。由于采用混杂模式的主机要处理大量的分组，所以它的负载必定很重，如果发现某个计算机的响应很慢，就可以怀疑它是工作于混杂模式。另外一种方法是使用错误的 MAC 地址和正确的 IP 地址向它发送 ping 数据包，如果它接收并应答了这个数据包，那一定是采用混杂模式进行通信的。

混杂模式通信被广泛地使用在恶意软件中，最初是为了获取根用户权限（Root Compromise），继而进行 ARP 欺骗（ARP Spoofing）。凡是进行 ARP 欺骗的计算机必定把网卡设置成了混杂模式，所以检测那些滥用混杂模式的计算机是很重要的。

9.8.2 网络嗅探器

嗅探器（Sniffer）就是采用混杂模式工作的协议分析器，可以用纯软件实现，运行在普通的计算机上；也可以做成硬件，用独立设备实现高效率的网络监控。Sniffer Network Analyzer 是美国网络联盟公司（Network Associates INC，NAI）的注册商标，然而许多采用类似技术的网络协议分析产品也可以叫做嗅探器。NAI 是电子商务和网络安全解决方案的主要供应商，它的产品除了 Sniffer Pro 之外，还有著名的防毒软件 McAfee。

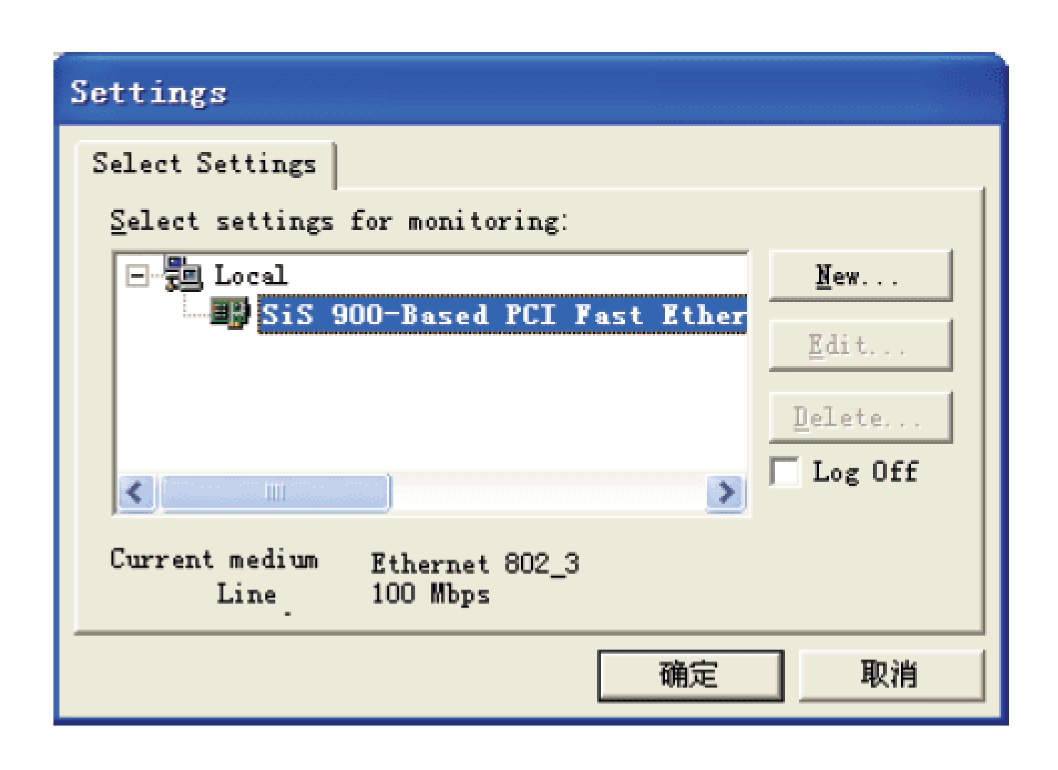


图 9-56 设置网卡

常用的 Sniffer Pro 网络分析器可以运行在各种 Windows 平台上。Sniffer 软件安装完成后在文件菜单中选择 Select Settings，就会出现图 9-56 所示的界

面，在这里可以选择用于监控的网卡，使其置于混杂模式。

9.8.3 Sniffer 软件的功能和使用方法

Sniffer Pro 主要包含 4 种功能组件：

（1）监视。实时解码并显示网络通信流中的数据。

（2）捕获。抓取网络中传输的数据包并保存在缓冲区或指定的文件中，供以后使用。

（3）分析。利用专家系统分析网络通信中潜在的问题，给出故障症状和诊断报告。

（4）显示。对捕获的数据包进行解码并以统计表或各种图形方式显示在桌面上。

网络监控是 Sniffer 的主要功能，其他功能都是为监控功能服务的。网络监控可以提供下列信息。

（1）负载统计数据，包括一段时间内传输的帧数、字节数、网络利用率、广播和组播分组计数等。

（2）出错统计数据，包括 CRC 错误、冲突碎片、超长帧、对准出错和冲突计数等。

（3）按照不同的底层协议进行统计的数据。

（4）应用程序的响应时间和有关统计数据。

（5）单个工作站或会话组通信量的统计数据。

（6）不同大小数据包的统计数据。

图 9-57 所示是 Sniffer 的系统界面，并且给出了监视菜单（Monitor）及其工具栏的解释。当 Sniffer 工作时，单击“主控板”按钮，可以显示网络利用率、数据包数/秒和错误数/秒三个计量表。这个窗口下面有如下三个选项（如图 9-58 所示）。

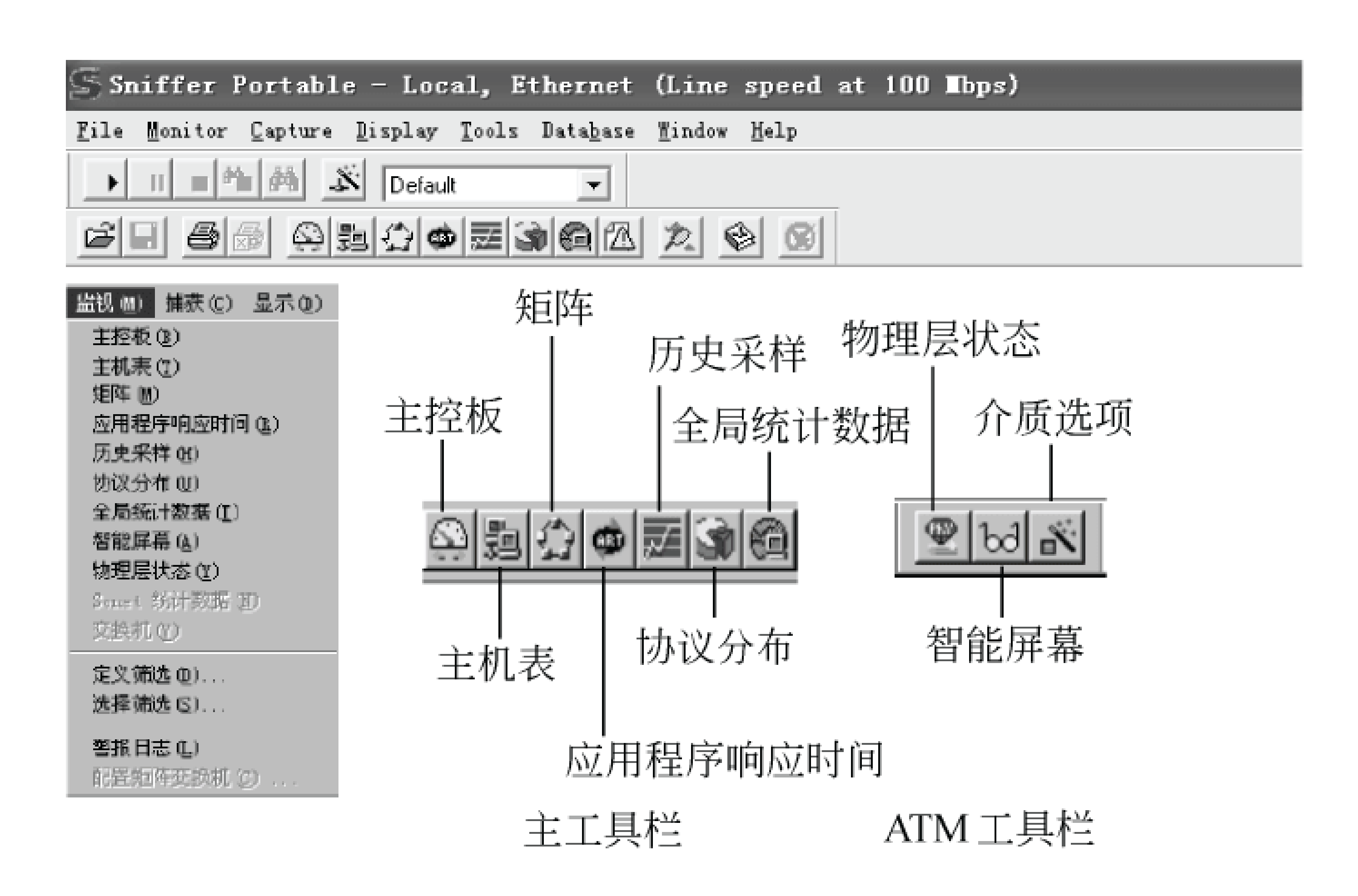
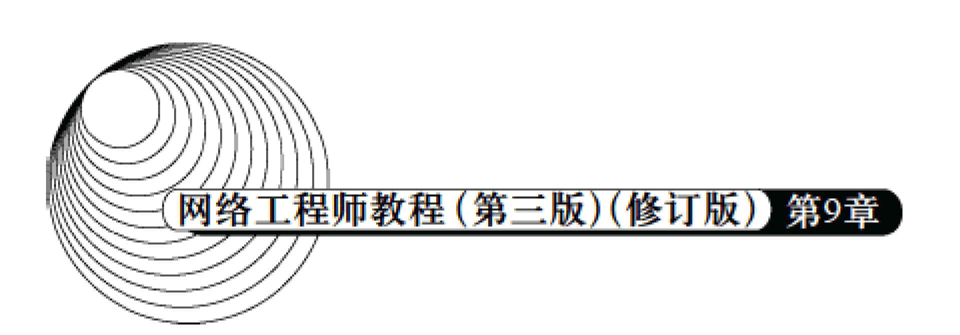


图 9-57 Sniffer 主菜单



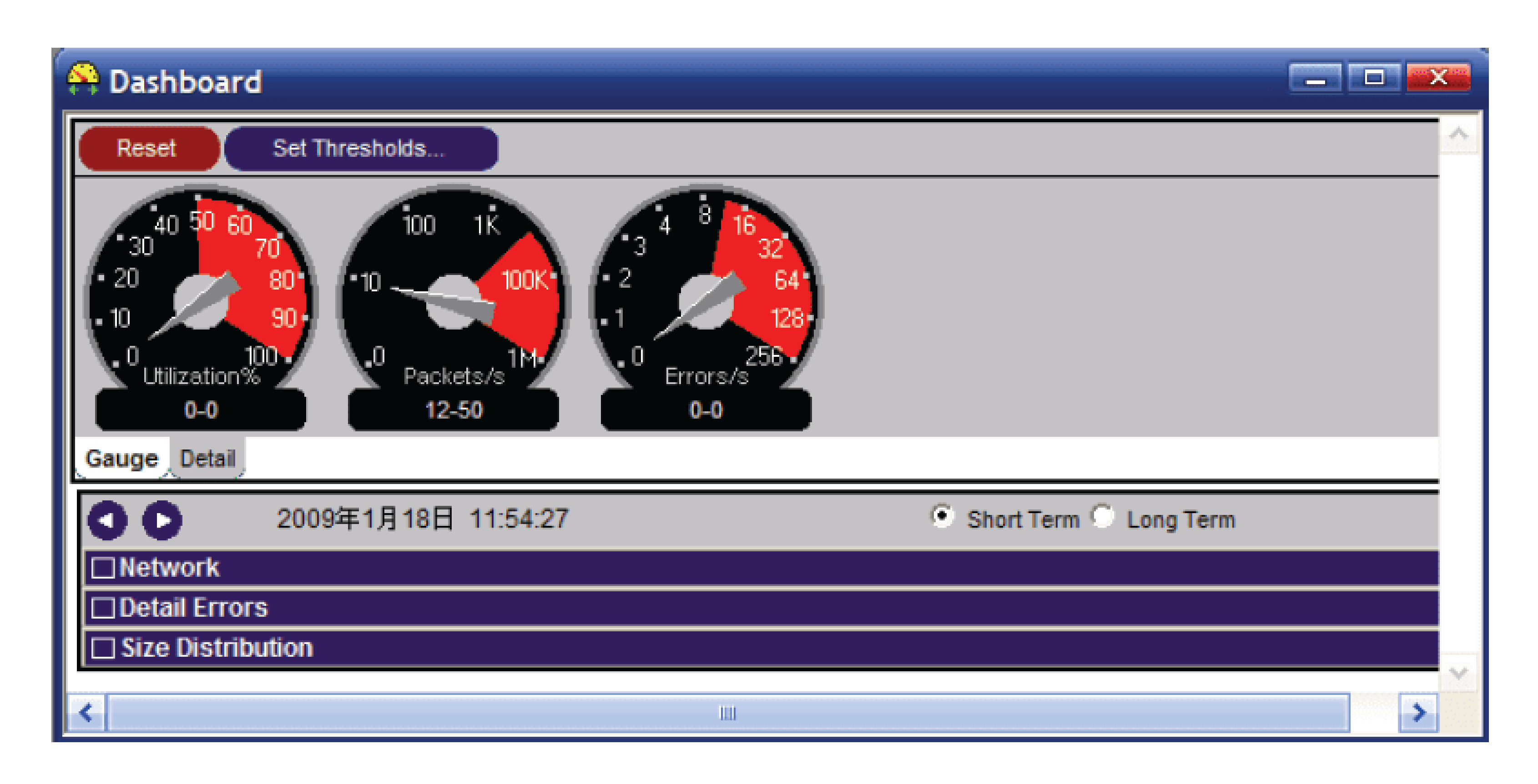


图 9-58　Sniffer 主控板

- Network：显示网络利用率等统计信息。
- Detail Errors：显示出错统计信息。
- Size Distribution：显示各种不同大小分组数的统计信息。

单击“主机表”按钮，可以显示通信最多的前 10 个主机的统计数据，如图 9-59 所示。单击“矩阵”按钮，可以显示主机之间进行会话的情况，如图 9-60 所示。其他按钮的使用是类似的，由于 GUI 界面直观易用，读者可以利用帮助信息熟悉 Sniffer 的使用方法。

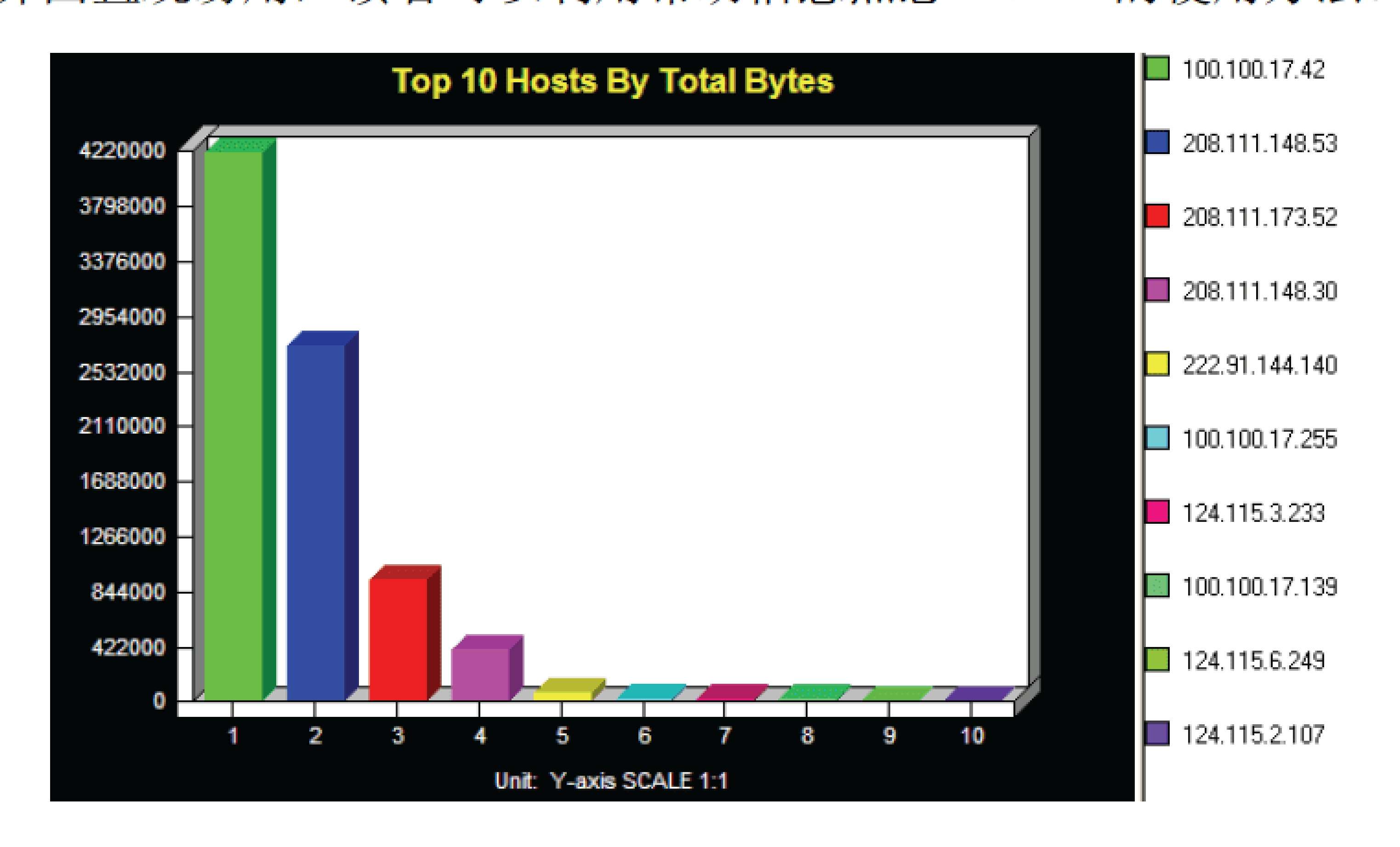


图 9-59　主机表

9.8.4　HP OpenView

HP OpenView 由多个功能套件组成，形成了一个集网络管理和系统管理为一体的完整系统。HP OpenView 包括以下套件。

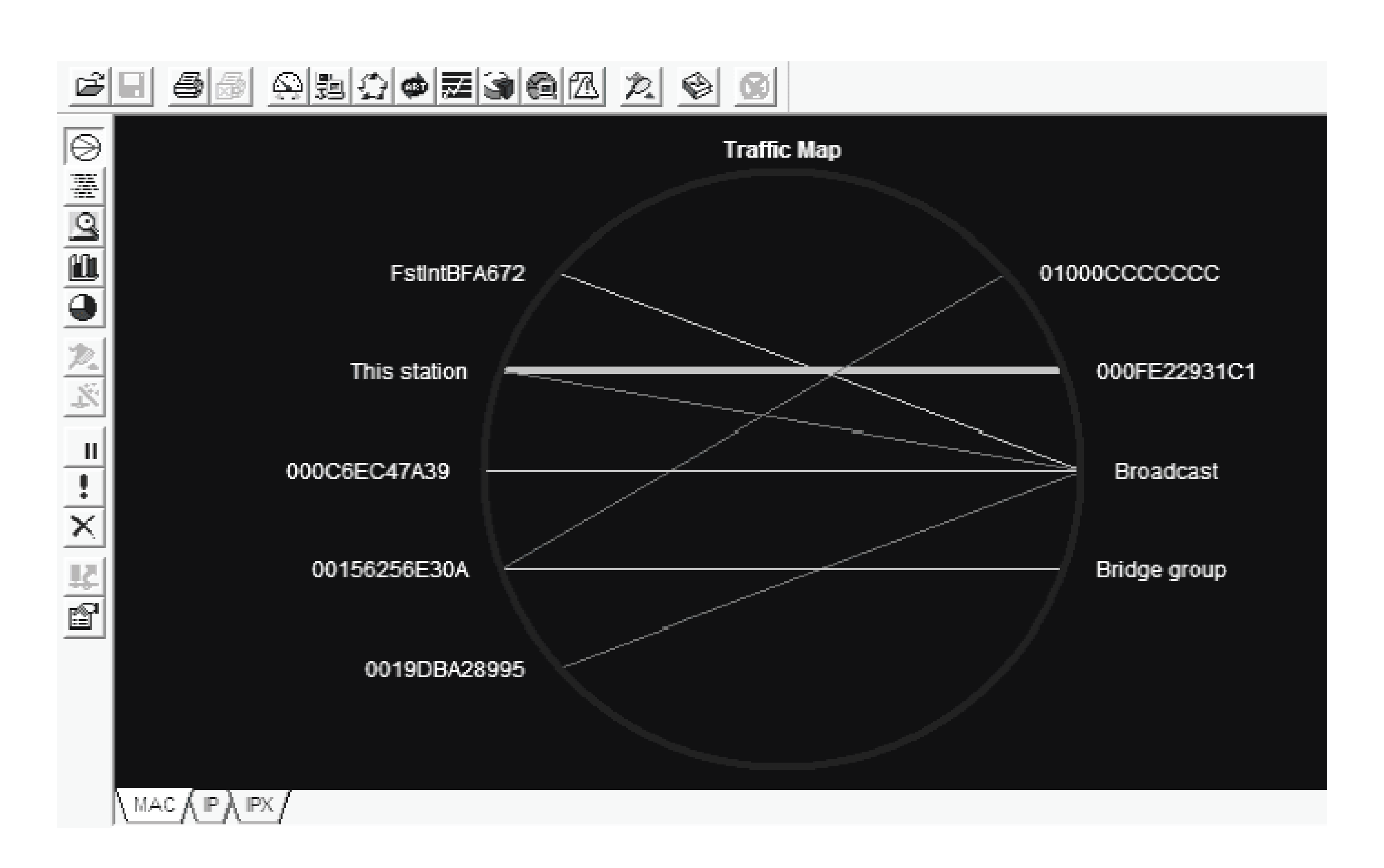


图 9-60 矩阵显示

- HP OpenView Operations：一体化的网络和系统管理平台，能支持数百个受控节点和数千个事件。
- HP OpenView Reporter：报告管理软件。为分布式 IT 环境提供灵活易用的报告管理解决方案，通过 Web 浏览器可以发布和访问各种管理报告。
- HP OpenView Performance：端到端的资源和性能管理软件。能收集、统计和记录来自应用、数据库、网络和操作系统的资源及性能测量数据。
- HP OpenView GlancePlus：实时诊断和监控软件。可以显示系统级、应用级和进程级的性能视图，诊断和识别系统运行中的问题和性能瓶颈。
- HP OpenView GlancePlus Pak 2000：全面管理系统可用性的综合性产品。在 GlancePlus Pak 的基础上增加了单一系统事件与可用性管理，可监控系统中的关键事件，使系统处于最佳性能状态。
- HP OpenView Database Pak 2000：服务器与数据库的性能管理软件。它提供强大的系统性能诊断功能，可以检测关键事件并采取修复措施，可提供 200 多种测量数据和 300 多种日志文件。

以上模块既相对独立，又可集成在一起，为企业提供高可用性的系统管理解决方案。

网络节点管理器

HP OpenView 最初是为网络管理设计的，其基础产品是网络节点管理器（Network Node Manager，NNM）。NNM 作为网络和系统管理的基础平台，可以与第三方管理应用集成在一起，形成强大的综合的网络管理环境。HP OpenView NNM 的主要功能特点分述如下。

（1）自动发现网络的拓扑结构，全面管理网络中的各种设备。NNM 能够自动发现网络节点，监测网络连接，生成和记录 TCP/IP 网络视图，通过不同彩色表示网络设备的运行状态。

发现和监控功能还可以探测广域网上的设备。通过 SNMP Data Presenter，用户可以查询网络的 SNMP 信息。

（2）具有管理大型、多节点网络的能力，可以适应多厂商设备、多操作系统的异构型环境。NNM 可以管理多达 1000 个以上的节点，能够适应地理上分布的网络环境。HP OpenView 是一种支持多厂商应用软件的管理平台，可以支持 21 种操作系统中的智能代理，包括 Windows、NetWare 和不同厂商的各种 UNIX 等。

（3）网络管理采用易于操作的图形界面。HP OpenView 采用图形用户界面，管理人员可以通过熟悉的点击、拖动、菜单选项等技术实现网络管理操作。使用 OpenView Windows 的窗格和缩放功能，在保持全网总图像的同时，可以将视点聚焦于重点子图的关键区域。

（4）与系统管理有机地集成在一起。HP OpenView 的网络管理产品可以紧密地结合到企业整体的资源与系统管理平台中，例如 HP OpenView Operation 中就内嵌了 NNM 模块。其他的网络管理模块都可以在 HP OpenView Operation 的操作平台上执行操作和显示数据。

（5）搜集到的信息可以进行有针对性地选择。NNM 对于所搜集到的信息具有简化功能，可提供发现过滤和拓扑过滤、图像过滤三种过滤方式，使管理人员可以根据需要选择要监控的对象，定制视图显示的内容和管理节点之间传输的信息。

（6）网络管理信息传输不会过多地占用网络资源。NNM 一方面可以对网络中的信息进行过滤，另一方面可以在本地进行网络故障的处理，只把故障事件和处理结果上报给上层控制台，从而减少了网络管理信息传输的通信流量。

（7）分布式的体系结构和远程管理操作。HP OpenView 的分布式解决方案便于协调管理人员的管辖范围，实现分层次的网络管理模式。NNM 能够通过 Web 界面访问网络拓扑和网管数据，在万维网的任何地点都可以进行远程管理操作。采用 HP OpenView Web Launcher 还可以在任何地点启动基于 Java 的 HP OpenView 应用，带有密码校验的登录过程确保了管理的安全性。

（8）故障的发现、显示与排除。NNM 能自动对网络进行监测，搜集网络中的故障和报警信息。NNM 采用事件关联技术，使得网管人员能够快速定位和排除故障。通过高级事件关联引擎把事件与高层次报警关联起来，可以立即发现网络故障的根本原因。

（9）与其他网管工具的集成。HP OpenView 提供了 SNMP 管理信息库的标准管理功能，用户还可以对 MIB 数据库进行扩展。HP OpenView 提供了标准的开发工具，用于开发可集成到管理平台上的应用软件。HP OpenView 已经被众多厂商作为其网络设备管理的平台软件。

（10）功能强大、简单易用的二次开发能力。HP OpenView 提供的各种应用开发包采用图形用户界面，无须具备特殊开发技巧就可以开发网管应用程序。HP OpenView 提供了基于 C 语言的 API，具有功能强大的可供调用的管理函数和公共服务，支持第三方合作伙伴开发多平台的、可扩展的分布式网络管理应用软件。

9.8.5 IBM Tivoli NetView

Tivoli NetView 是 IBM 公司的网络管理工具，能够提供整个网络环境的完整视图，实现对网络产品的管理。它采用 SNMP 协议对网络上的设备进行实时的监控，对网络中发生的故障进行报警，从而减少了系统管理的难度和管理工作量。

IBM Tivoli NetView 网络管理解决方案可以实现的功能主要包括：

（1）网络拓扑管理。NetView 能够自动发现联网的 IP 节点，包括路由器、交换机、服务器和 PC 等，并自动生成拓扑连接。NetView 还可以按照地理位置对网络拓扑图形进行定制，使之与实际的网络结构更加吻合。图 9-61 是 Tivoli 网络管理拓扑显示界面。

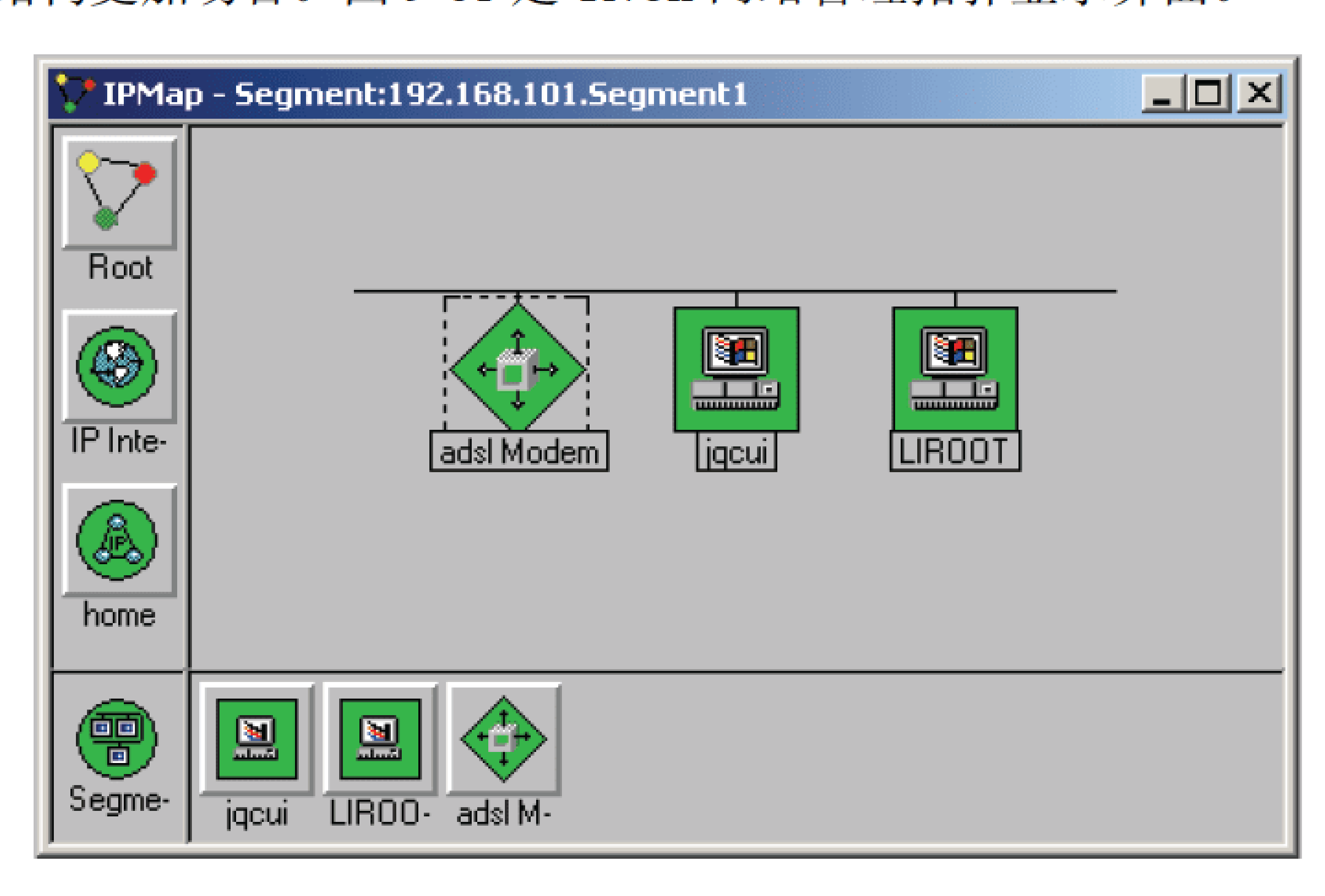


图 9-61 Tivoli 网络管理拓扑显示界面

NetView 提供的 SmartSet 功能可以将具有相同属性的管理对象组成一个集合，例如用户可以把重要的路由器放在一起作为一个集合，进行统一的管理设置。SmartSet 甚至不需要手工加入对象，管理员只需设置加入集合的条件，SmartSet 就能够动态发现符合条件的设备并自动加入集合视图，从而为管理员提供了很大的便利。

（2）网络故障管理。网络故障管理是网络管理的核心。NetView 的图形化网络拓扑结构可以迅速发现出现故障的资源，并帮助管理员分析故障原因。当网络中的设备出现故障、死机或链路中断时，NetView 会及时在屏幕上显示报警信号，便于网络管理人员进行诊断，并排除故障。

（3）网络性能管理。NetView 的 SnmpCollect 功能可以自动采集重要的网络性能数据，例如 IP 流量、带宽利用率、出错包数量、丢弃包数量和 SNMP 流量等。通过设置各种参数的阈值，NetView 能够自动发出报警信号，或自动运行已定义的管理操作。NetView 可以用图形的

方式显示网络性能数据的变化情况，或者将管理数据存放在关系数据库中，以便于以后进行检索和分析。图 9-62 所示为网络性能分析视图。

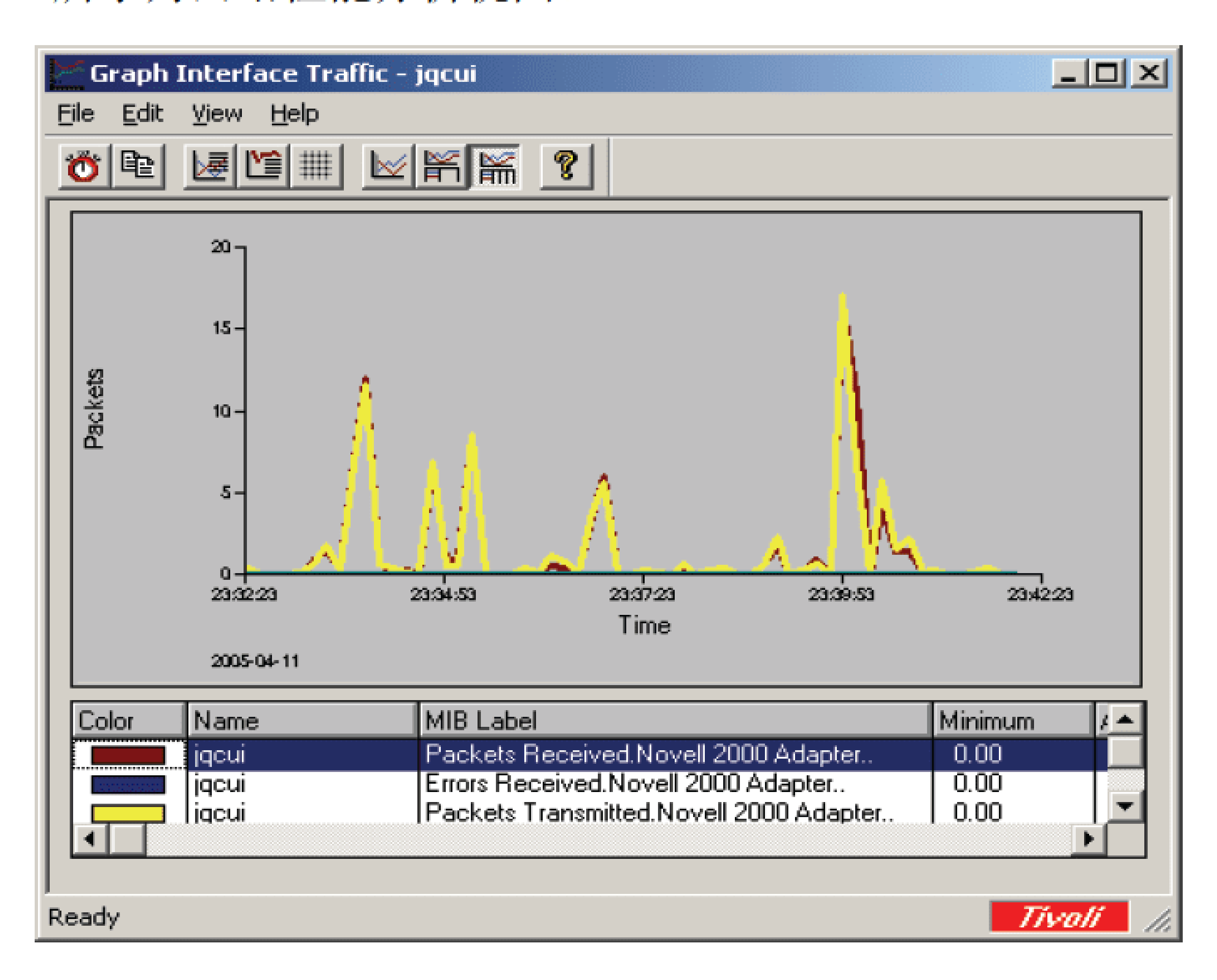


图 9-62 网络性能分析监控显示

Tivoli 数据仓库为网络性能管理提供集中的历史统计和报表分析，能够帮助管理人员从大量数据中及时发掘出可用于判断网络运行状态的数据，能够生成各种报表和图形化的分析报告。

（4）网络设备管理。Tivoli NetView 是使用最广泛的网络管理平台之一，支持业界标准 API，能够与主要网络设备厂商的设备管理软件（如 Cisco Works、Nortel Optivity 和 3com Transcend 等）方便地进行集成。

（5）管理权限分配。NetView 可以为管理员定义不同的管理角色，不同的管理角色可以被授权管理不同地域范围的设备，没有权限管理的设备不会出现在网络拓扑视图中。

（6）Web 管理功能。NetView 通过 Web 控制台实现了分布式的网络管理。NetView Web 控制台为用户提供了一个灵活、可配置的环境，便于用户远程访问网络设备、浏览交换机的端口、检查路由器的工作状态、查看 MAC 地址等。

（7）支持 MPLS 管理功能。NetView 7.1 支持对多协议标记交换设备的识别，并能对有关 MPLS 的数据进行查询，可以管理 LSR 设备。

（8）交换机的故障定位。IBM Tivoli Switch Analyzer 提供了第二层交换设备的发现功能，能够识别包括第二层和第三层交换设备在内的各种设备之间的关系。正确地关联分析可以区分不同的设备，无论是 IP 寻址的端口，还是第二层交换机上非 IP 寻址的端口、板卡或插件。

9.8.6 CiscoWorks for Windows

CiscoWorks for Windows 是基于 Web 的网络管理解决方案，主要应用于中小型企业网络，提供了一套功能强大、价格低廉且易于使用的监控和配置工具，用于管理 Cisco 的交换机、路由器、集线器、防火墙和访问服务器等设备。使用 Ipswitch 公司的 WhatsUp Gold 工具，还可管理网络打印机、工作站、服务器和其他网络设备。CiscoWorks for Windows 中包含下列组件。

（1）CiscoView。CiscoView 可以提供设备前后面板的视图，能够以不同颜色动态地显示设备状态，并提供对特定设备组件的诊断和配置功能。CiscoView 启动后可以从设备列表中选择要监视的设备。如果要监视的设备不在设备列表中，则直接输入设备 IP 地址。选择了一个设备之后，将出现有关该设备信息的界面，如图 9-63 所示。

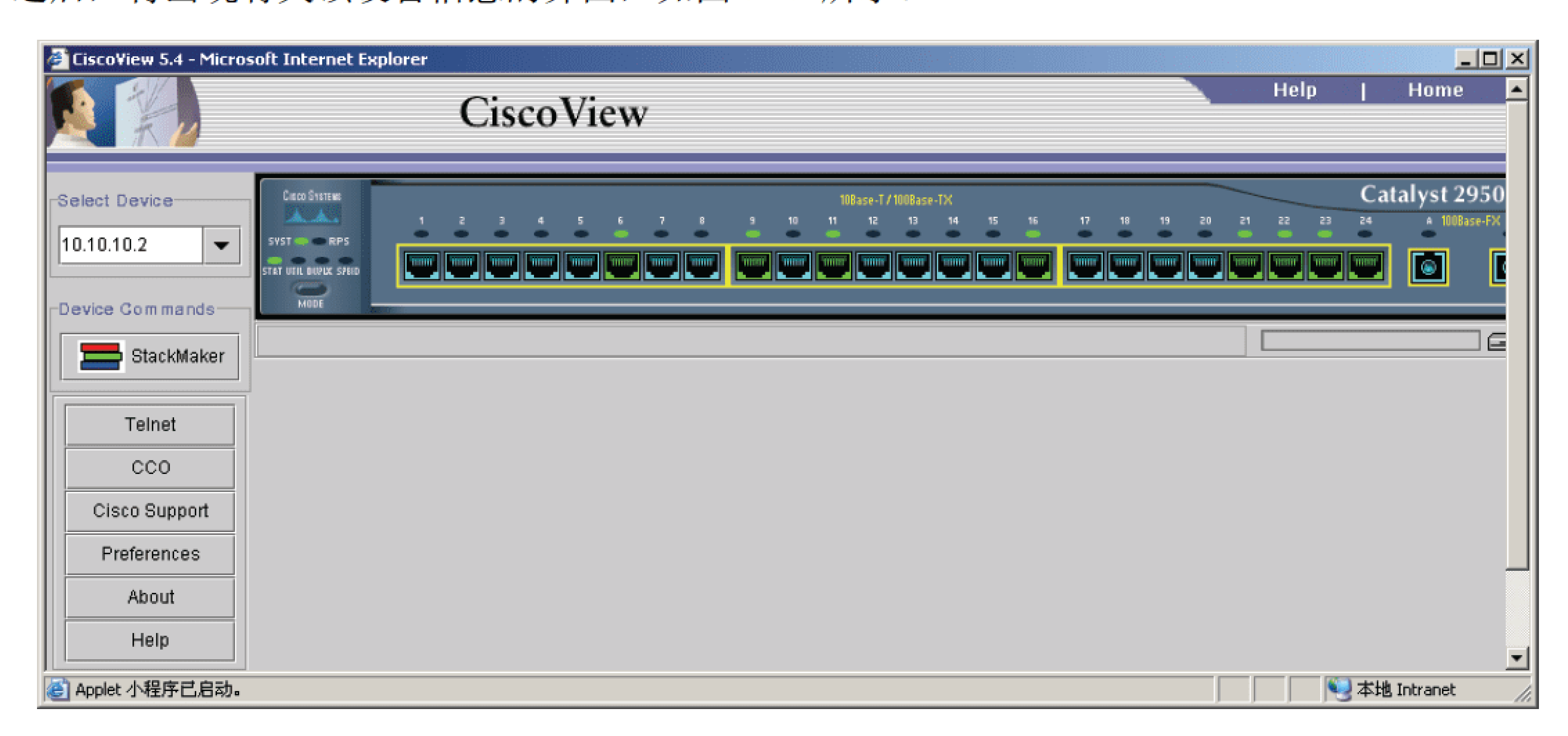


图 9-63　CiscoView 界面

（2）WhatsUp Gold。WhatsUp Gold 是一种基于 SNMP 的图形化网络管理工具，可以通过自动或手工创建网络拓扑结构图管理整个企业网络，支持监视多个设备，具有网络搜索、拓扑发现、性能监测和警报追踪等功能。WhatsUp Gold 的界面如图 9-64 所示。

（3）门限管理。门限管理器（Threshold Manager）能够在支持 RMON 的 Cisco 设备上设置门限值并提取事件信息，以增强排除网络故障的能力。使用 Threshold Manager 之前，必须建立门限模板。Cisco 公司提供了一些预定义的模板，用户也可以定义自己的模板。Threshold Manager 管理界面如图 9-65 所示。

在图 9-65 中，Event Log 窗口以表格的方式显示越界事件信息，并把 RMON 日志记录保存在被管理设备上；Device Thresholds 窗口用来设置和显示阈值；Templates 窗口用来显示所有默

认的或用户定制的模板，也可以建立新的模板；Trap Receivers 窗口可以添加或删除接收陷入事件的管理站点；Preferences 窗口则用来设置 Threshold Manager 的属性。

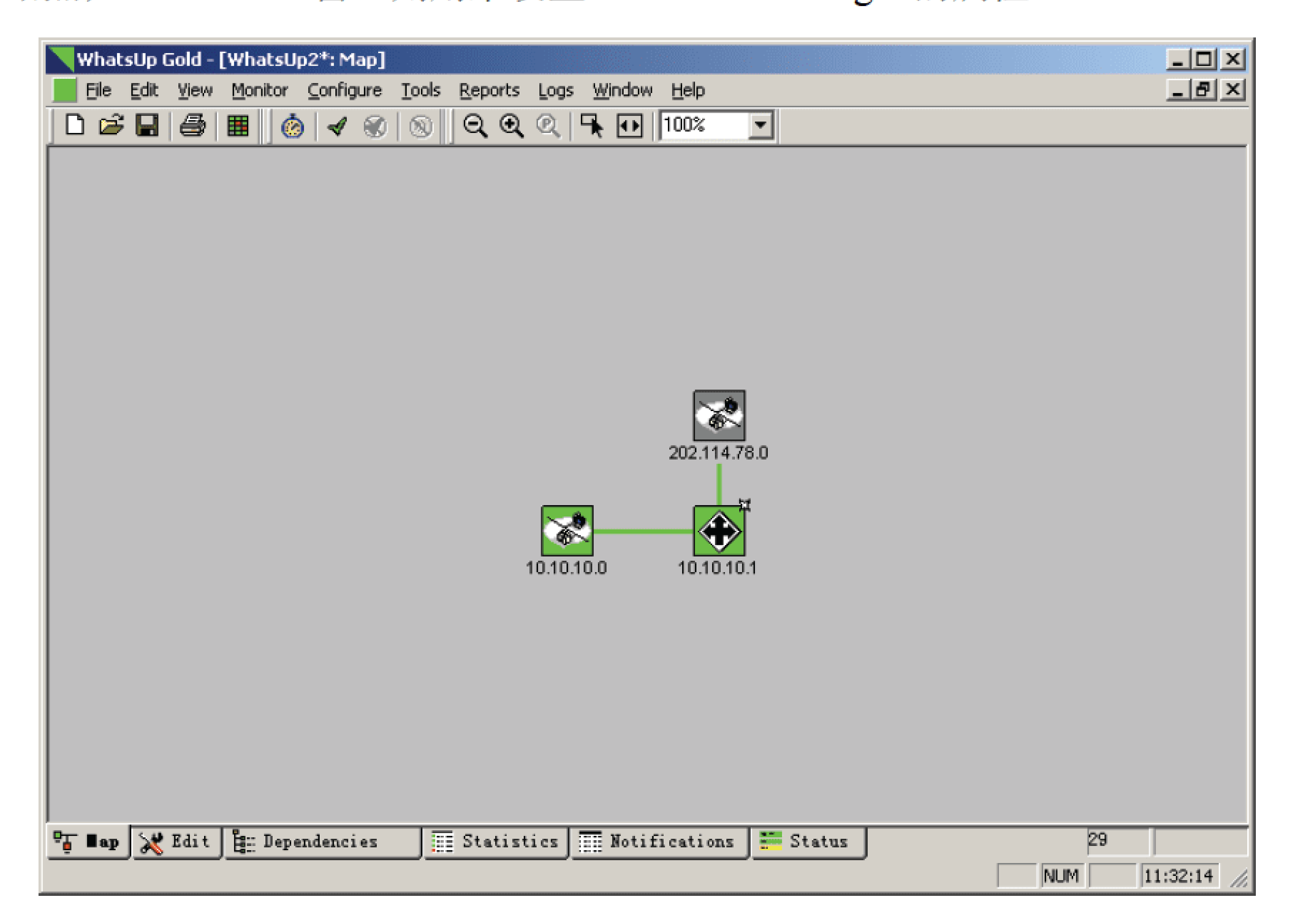


图 9-64 WhatsUp Gold 用户界面

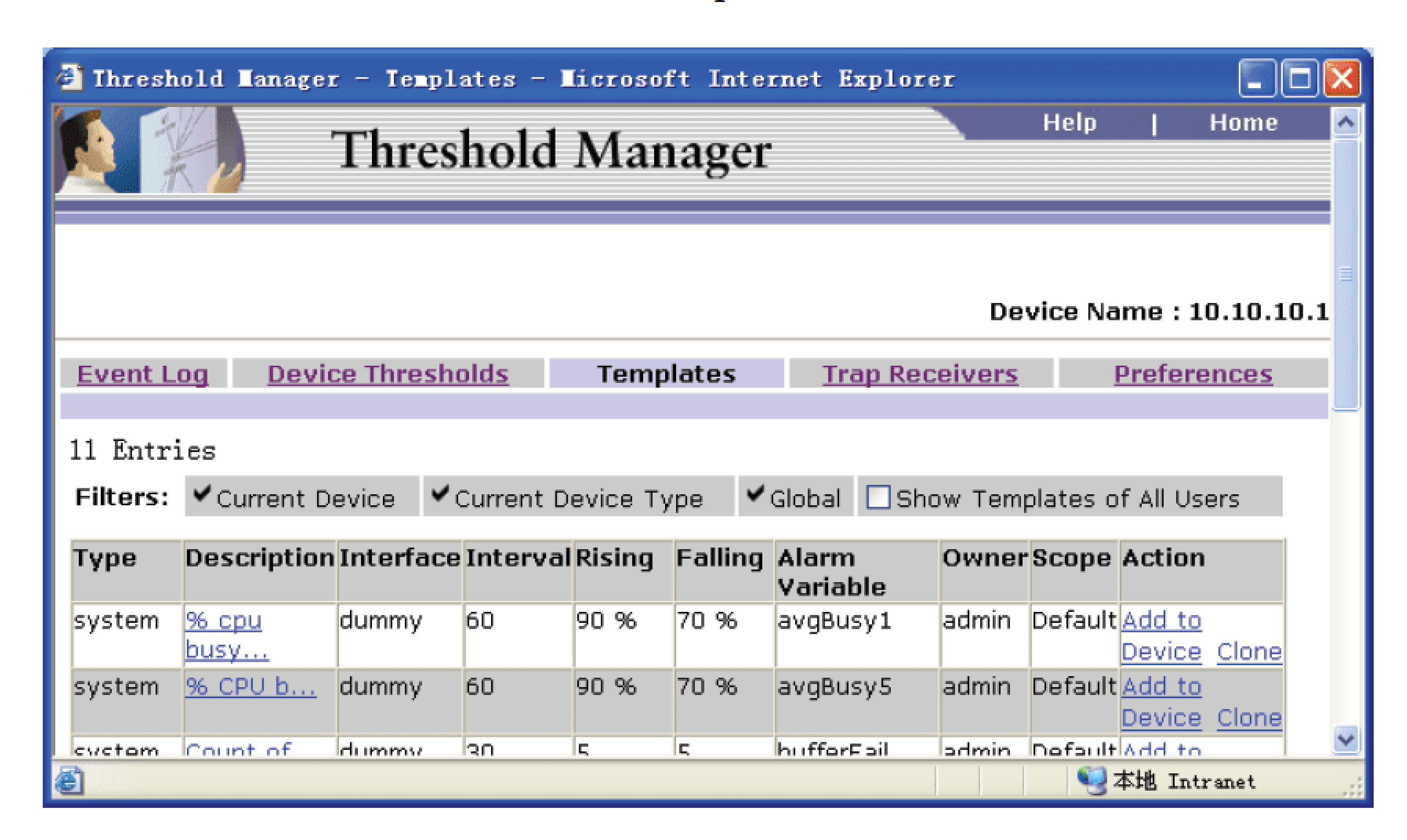


图 9-65 Threshold Manager 管理界面

（4）Show Commands。Show Commands 使得用户不必记住各个设备的命令行语法，使用

Web 浏览器进行简单操作就可以获取设备的系统信息和协议信息。Show Commands 在 Web 页面的左边以树型结构显示了设备所支持的命令列表，如图 9-66 所示。当用户选择了一个命令后，Show Commands 将执行所选择的命令，并显示命令行的输出信息。

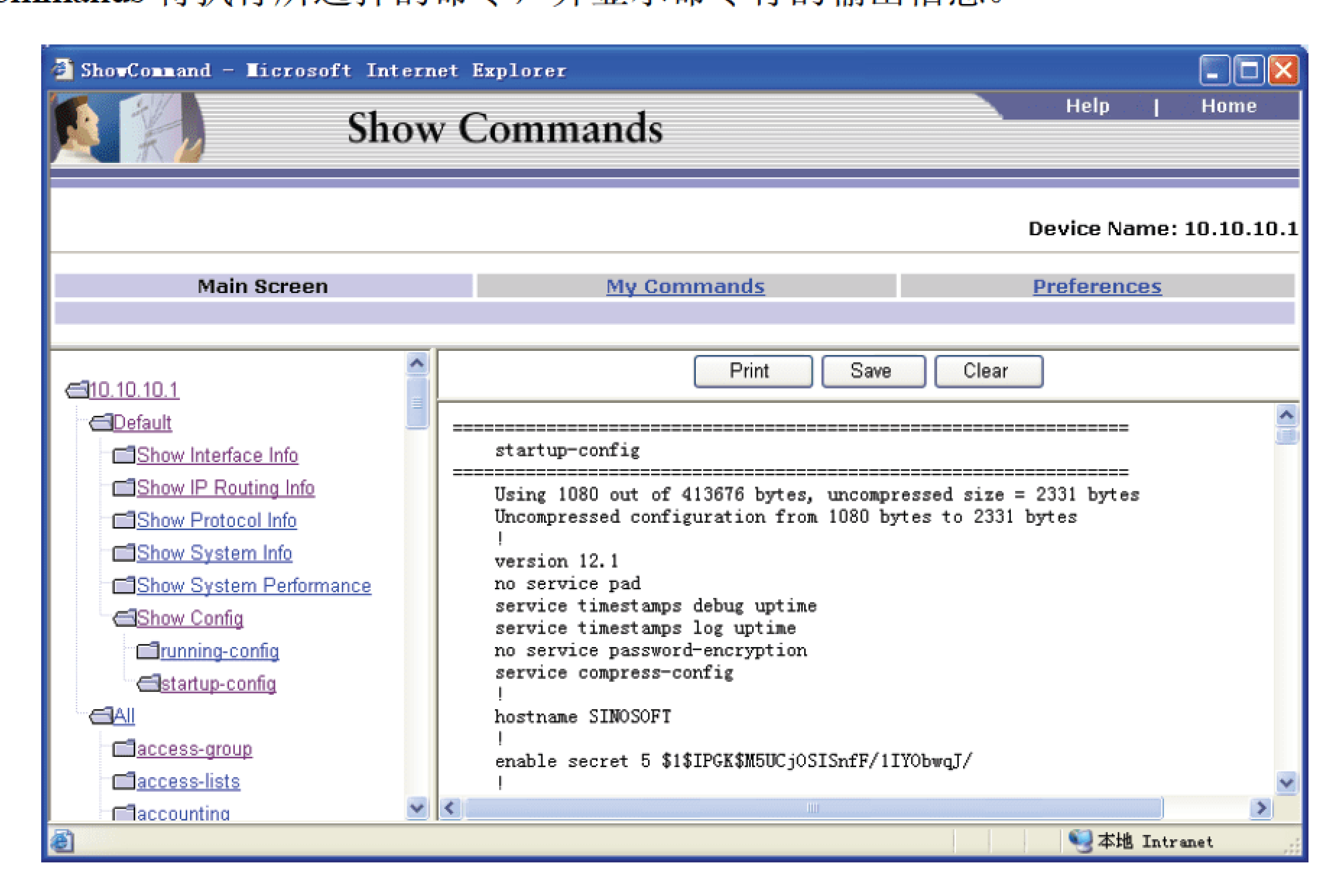


图 9-66　Show Commands 操作界面

9.9　网络存储技术

9.9.1　廉价磁盘冗余阵列

廉价磁盘冗余阵列（Redundant Arrays of Inexpensive Disk，RAID）是美国加利福尼亚大学伯克莱分校在 1987 年提出的，它是利用一台磁盘阵列控制器管理一组（几台到几十台）磁盘驱动器，组成一个可靠的、快速的大容量磁盘系统。

冗余磁盘阵列技术最初的研制目的是为了组合小型的廉价磁盘来代替大容量的昂贵磁盘，以降低大批量数据存储的费用，同时也希望采用冗余技术提高磁盘数据的可靠性，并能适当提升数据传输的速率。RAID 有时也被称为独立磁盘冗余阵列（Redundant Array of Independent Disk），以强调其可作为一台虚拟的大容量硬盘使用的特点。

RAID 的重要特性是所谓的 EDAP（Extended Data Availability and Protection）概念，强调了这种系统的可扩充性和容错机制。RAID 在不停机的情况下可支持以下功能。

（1）自动检测硬盘故障。

（2）重建硬盘的坏道信息。

（3）硬盘热备份。

（4）硬盘热替换。

（5）扩充硬盘容量。

过去 RAID 一直作为高档 SCSI 硬盘的配套技术在高档服务器中使用，近年来随着技术的发展和产品成本的下降，IDE 硬盘性能有了很大提升，加之 RAID 芯片的普及，使得 RAID 也逐渐应用到个人计算机上。

RAID 规范包含 RAID 0～RAID 7 多个等级，它们的技术特点各不相同，目前投入商业应用的有下列几种。

1．RAID 0

RAID 0 需要两个以上硬盘驱动器，每个磁盘划分为不同的区块，如图 9-67 所示。数据按区块 A1、A2、A3、A4…的顺序存储，数据访问采用交叉存取、并行传输的方式。将数据分布在不同驱动器上，可以提高传输速度，平衡驱动器的负载。但这种系统没有差错控制措施，如果一个盘上的数据出现错误，其他盘上的数据也无用了。RAID 0 不能用于对数据稳定性要求较高的场合。如果进行图像编辑，或其他要求传输速度比较高的场合，使用 RAID 0 比较合适。在所有级别中，RAID 0 的速度是最快的。

2．RAID 1

具有磁盘镜像功能，可利用并行读写特性，将数据分块并同时写入主磁盘和镜像盘，磁盘容量的利用率只有 50%，它是以牺牲磁盘容量为代价换取可靠性的提高。在图 9-68 中，磁盘 1 是主磁盘，磁盘 2 是镜像盘。

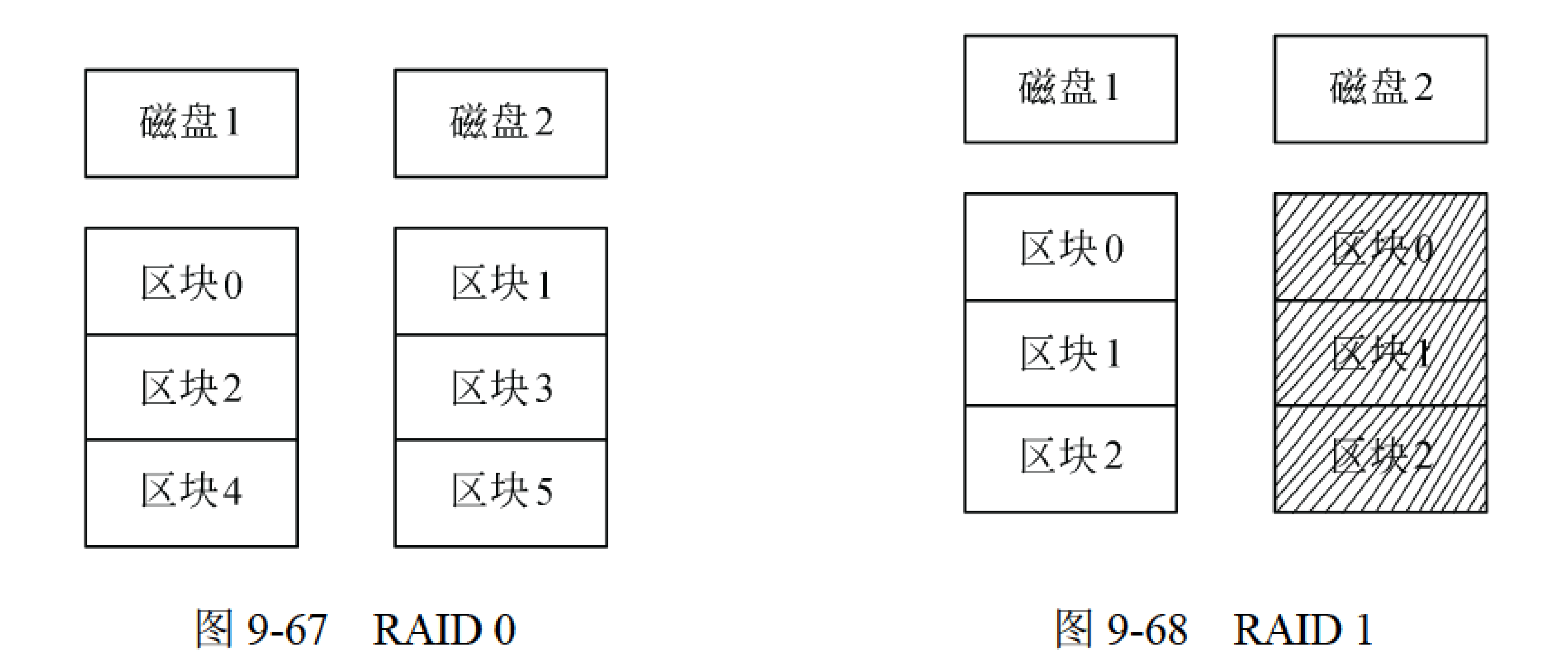


图 9-67　RAID 0　　　图 9-68　RAID 1

RAID 1 控制器能够同时对两个盘进行读写操作，通过镜像技术提高系统的容错能力。当主硬盘损坏时，镜像硬盘就可以代替主硬盘工作，镜像硬盘相当于一个备份盘，这种硬盘控制模式的安全性是非常高的。RAID 1 的差错校验功能对系统的处理能力有很大影响，通常的RAID功能由软件实现，在服务器负载比较重时会影响其工作效率。当系统需要极高的可靠性时，如进行数据统计，使用 RAID 1 比较合适。RAID 1 技术支持热替换，即在不断电的情况下对故障磁盘进行更换，更换完毕后只要从镜像盘上恢复数据即可。

3. RAID 2 和 RAID 3

RAID 2 与 RAID 3 类似，两者都是将数据分块存储在不同的硬盘上实现多模块交叉存取，并在数据访问时提供差错校验功能。RAID 2 使用海明码进行差错校验，需要单独的磁盘存放校验与恢复信息。RAID 2 的实现技术代价昂贵，在商业环境中很少使用。

RAID 3 采用奇偶校验方式，只能查错不能纠错。这种技术需要三个以上的驱动器，一个磁盘专门存放奇偶校验码，其他磁盘作为数据盘实现多模块交叉存取，如图 9-69 所示。RAID 3 访问数据时一次处理一个区块，这样可以提高读取和写入的速度，奇偶校验码在写入数据时产生并保存在校验盘上。RAID 3 主要用于图形图像处理等要求吞吐率比较高的场合，对于大量的连续数据可提供良好的传输速率，但对于随机数据，奇偶校验盘会成为写操作的瓶颈。利用单独的奇偶校验盘来保护数据使磁盘的利用率提高到（$n-1$）/n。

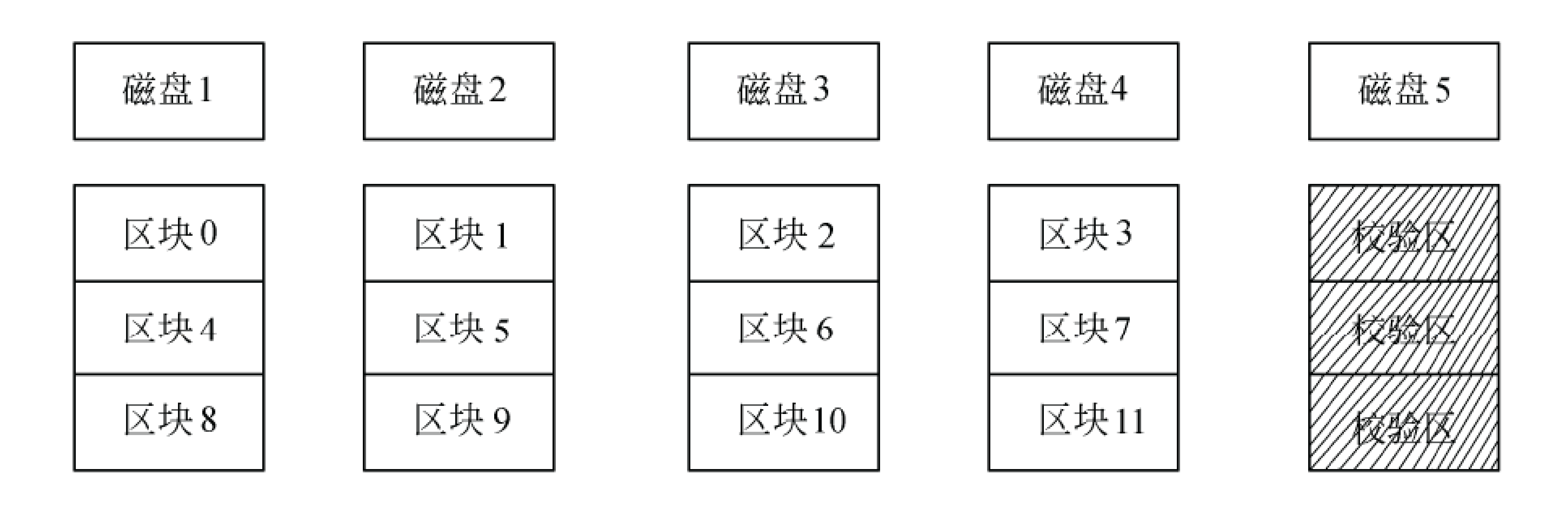


图 9-69 RAID 3

4. RAID 5

这是一种分布式奇偶校验的独立磁盘结构。与 RAID 3 不同的地方是，用来进行纠错的校验信息分布在各个数据盘上，没有专门的校验盘，图 9-70 中的 P01 表示区块 0 和区块 1 按位异或运算后得到的校验和，其余类推。这种校验方式允许任何一台磁盘机损坏，例如磁盘 3 坏了，则可以用区块 0 和区块 1 进行异或运算重新得到 P01，用 P23 和区块 3 进行异或运算重新得到

区块 3，依此类推。

RAID 5 的读出效率很高，写入效率一般，对区块式的聚集访问效率不错。由于奇偶校验码分布在不同的磁盘上，允许单个磁盘出错，所以提高了可靠性，也提高了磁盘的利用率。但是它对数据传输的并行性解决得不好，而且控制器的设计也相当复杂。对于 RAID 5 来说，大部分数据传输只对一块磁盘操作，可进行并行访问。

磁盘 1	磁盘 2	磁盘 3
区块 0	区块 1	P01
P23	区块 2	区块 3
区块 5	P45	区块 4
区块 6	区块 7	P67

图 9-70　RAID 5

5．RAID 0+1

正如其名字所暗示的一样，RAID 0+1 是 RAID 0 和 RAID 1 的组合形式，也称为 RAID 10。以 4 个磁盘组成的 RAID 0+1 为例，其数据存储方式如图 9-71 所示。RAID 0+1 是存储性能和数据安全兼顾的方案。它在提供与 RAID 1 同样的数据安全保障的同时，也提供了与 RAID 0 近似的访问速率。

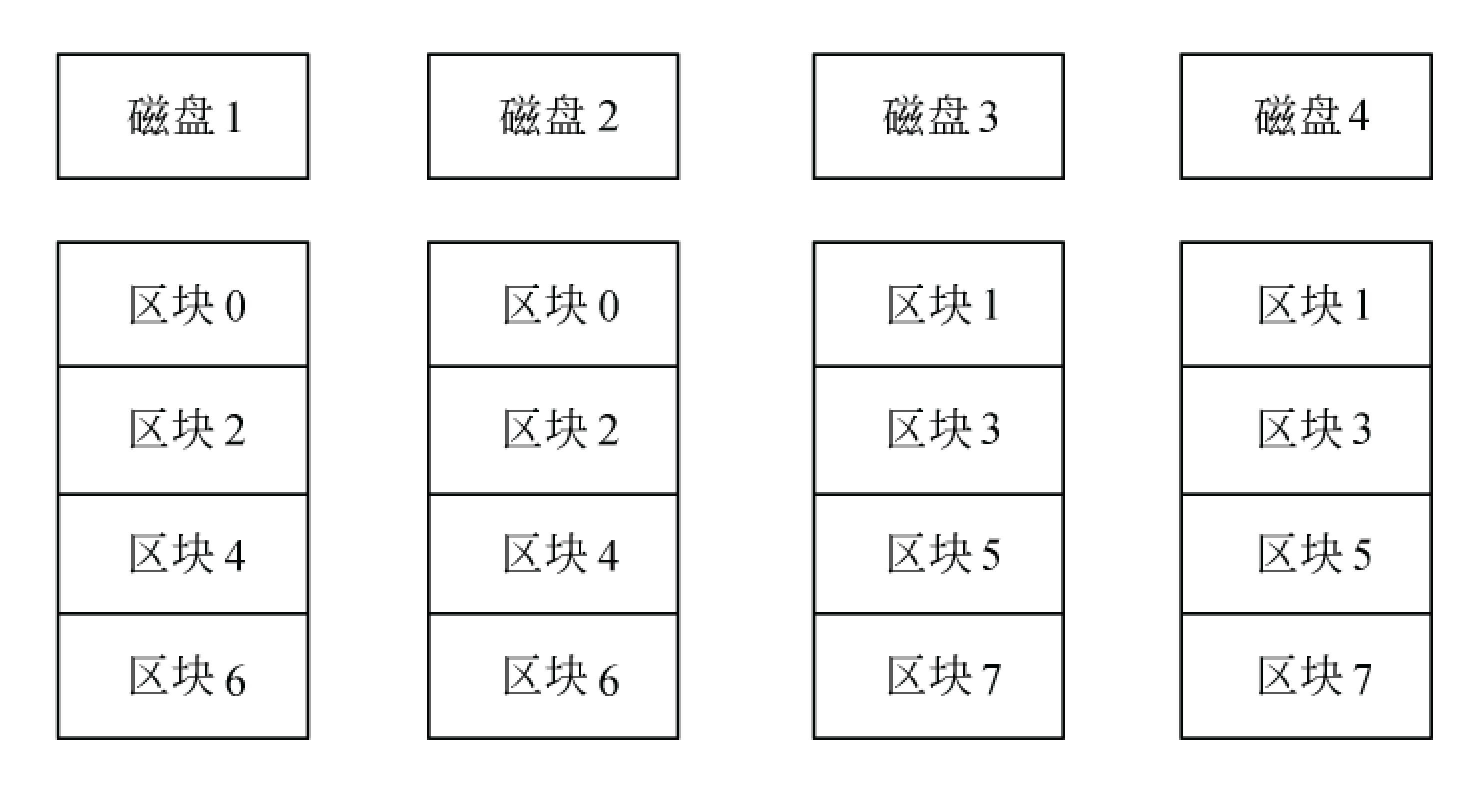


图 9-71　RAID 0+1

由于 RAID 0+1 通过数据的 100%备份提供数据安全保障，因此 RAID 0+1 的磁盘空间利用率与 RAID 1 相同，存储成本很高。

RAID 0+1 的特点使其特别适用于既有大量数据需要存取，同时又对数据安全性要求严格的领域，例如银行、金融、商业超市、仓储库房和各种档案管理等。

6．JBOD 模式

JBOD 代表 Just a Bunch of Drives，它是在逻辑上将几个物理磁盘连接起来， 组成一个大的逻辑磁盘。JBOD 不提供容错，其容量等于所有磁盘容量的总和。严格意义上说，JBOD 不

属于 RAID 的范围，不过现在很多 IDE RAID 控制芯片都带有这种模式。JBOD 就是简单的硬盘容量叠加，但系统处理时并没有采用并行的方式，写入数据的时候是先写一块硬盘，写满了再写第二块硬盘。

实际应用中最常见的是 RAID 0、RAID 1、RAID 5 和 RAID 10。由于在大多数场合，RAID 5 包含了 RAID 2～4 的优点，所以 RAID 2～4 基本退出市场，一般认为 RAID 2～4 只用于 RAID 的开发研究领域。

9.9.2　网络存储

基于 Windows、Linux 和 UNIX 等操作系统的服务器称为开放系统。开放系统的数据存储方式分为内置存储和外挂存储两种，而外挂存储又根据连接的方式分为直连式存储和网络化存储，目前应用的网络化存储方式有两种，即网络接入存储和存储区域网络，如图 9-72 所示。下面介绍开放系统的外挂存储方式。

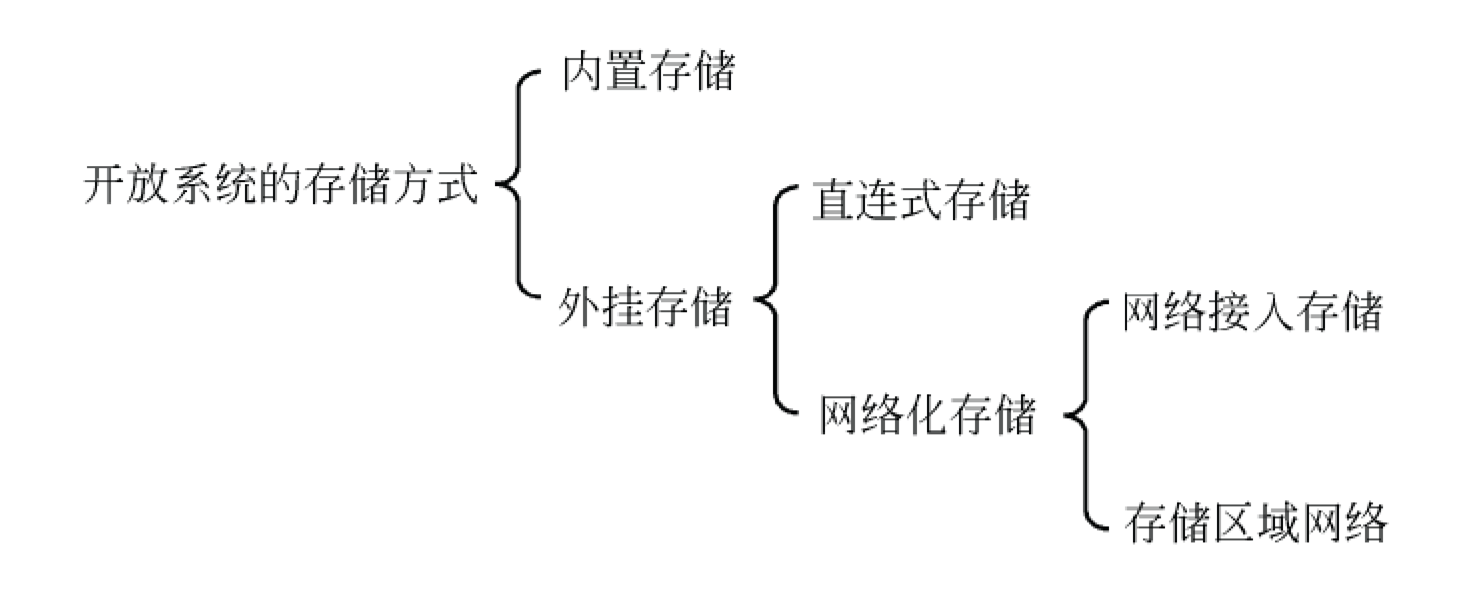


图 9-72　存储系统的分类

1. 直连式存储

开放系统的直连式存储（Direct-Attached Storage，DAS）如图 9-73 所示，即在服务器上外挂了一组大容量硬盘，存储设备与服务器主机之间采用 SCSI 通道连接，带宽为 10Mbps、20Mbps、40Mbps 和 80Mbps 等。

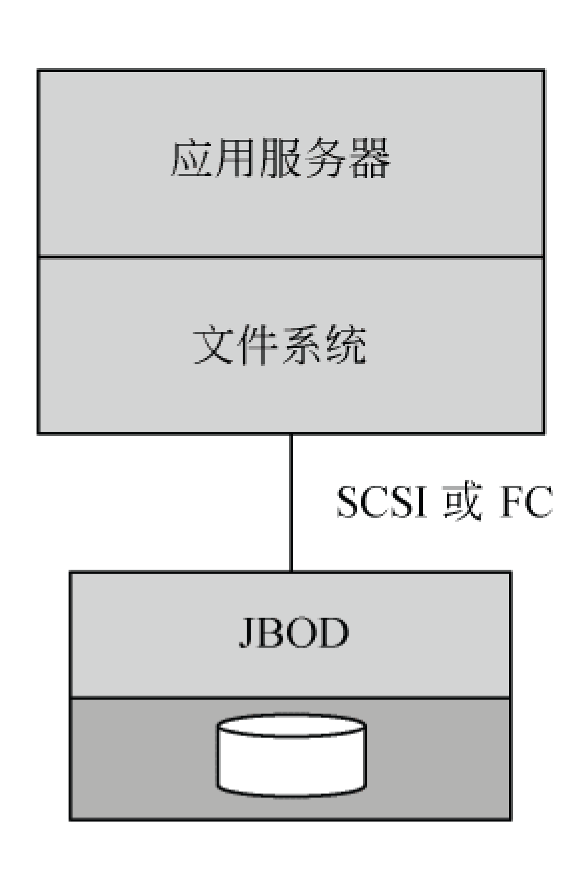
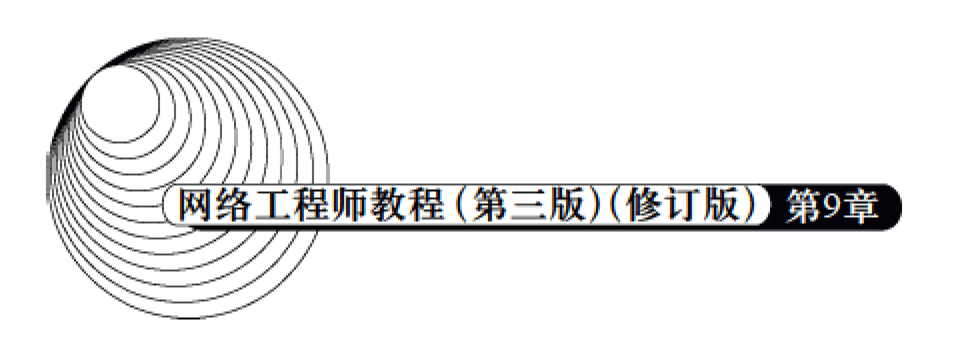


图 9-73　DAS

直连式存储直接将存储设备连接到服务器上，这种方法难以扩展存储容量，而且不支持数据容错功能，当服务器出现异常时，会造成数据丢失。

随着服务器 CPU 处理能力的不断增强，磁盘存储空间越来越大，硬盘数量越来越多，SCSI 通道将会成为 I/O 瓶颈。同时，



由于服务器主机的 SCSI ID 资源有限，能够建立的 SCSI 通道连接也有限。无论存储阵列或是服务器主机的扩展，都会造成系统的停机，从而给企业带来经济损失，对于银行、电信和传媒等需要 7×24 小时服务的行业，这是不可接受的。

DAS 已经有近 40 年的使用历史，目前正在让位于日渐兴盛的网络化存储。

2. 网络接入存储

网络化存储的出现适应了网络成为主要信息处理平台的发展趋势，它分摊了数据处理和存储管理的功能，计算机负责数据处理，而存储子系统负责数据的存储和管理。网络化存储能够提供灵活的解决方案，利用专用的存储子系统可以实现以下功能。

（1）在多个存储子系统之间合理地分配存储任务。

（2）在多个存储位置之间实现可靠的数据传输。

（3）实现可靠的数据保护和数据恢复功能。

（4）实现多个主机系统对数据的并行访问。

网络接入存储（Network Attached Storage，NAS）是将存储设备连接到现有的网络上，来提供数据存储和文件访问服务的设备。NAS服务器是在专用主机上安装简化了的瘦操作系统（只具有访问权限控制、数据保护和恢复等功能）的文件服务器。NAS 服务器内置了与网络连接所需要的协议，可以直接联网，具有权限的用户都可以通过网络来访问 NAS 服务器中的文件。NAS 服务器直接连接磁盘阵列，它具备磁盘阵列的所有特征：高容量、高效能、高可靠性。NAS 是真正即插即用的产品，物理位置灵活，可放置在工作组内，也可放在其他地点。用户之所以选择 NAS 解决方案，原因是 NAS 价格合理、便于管理、灵活且能实现文件共享。

典型的 NAS 都连接到普通的以太网上，提供预先配置好的磁盘容量和存储管理软件，成为完备的网络存储解决方案，如图 9-74 所示。

3. 存储区域网络

存储区域网络（Storage Area Network，SAN）是一种连接存储设备和存储管理子系统的专用网络，专门提供数据存储和管理功能。SAN 可以被看作是负责数据传输的后端网络，而前端网络（或称为数据网络）则负责正常的 TCP/IP 传输。也可以把 SAN 看作是通过特定的互连方式连接的若干台存储服务器组成的单独的数据网络，提供企业级的数据存储服务，其拓扑结构如图 9-75 所示。

SAN 是一种特殊的高速网络，采用光纤通道（Fibre Channel）实现互连，通过光纤通道交换机连接存储阵列和文件服务器主机。SAN 不仅可以提供大容量的存储数据，而且地域上可以分散部署，从而缓解了大量数据传输对于局域网通信的影响。SAN 的结构使得文件服务器可以连接到任何存储阵列，不管数据存放在哪里，服务器都可直接访问需要的数据。

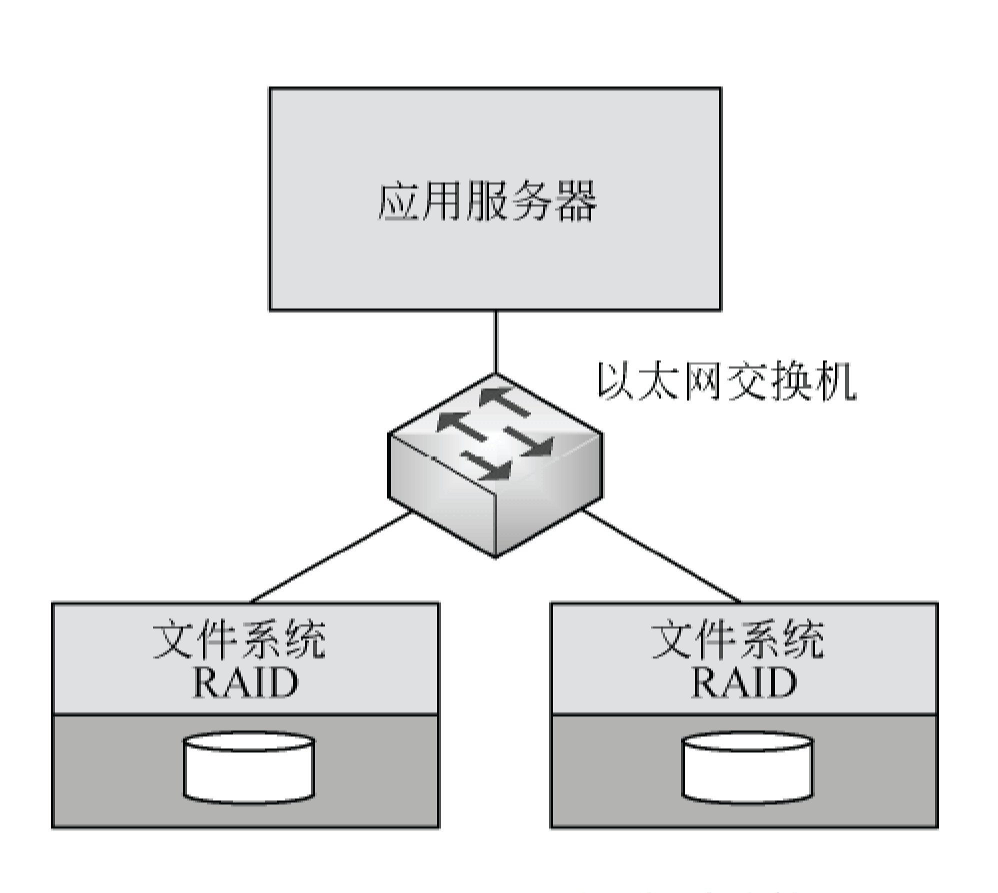


图 9-74 NAS 的体系结构

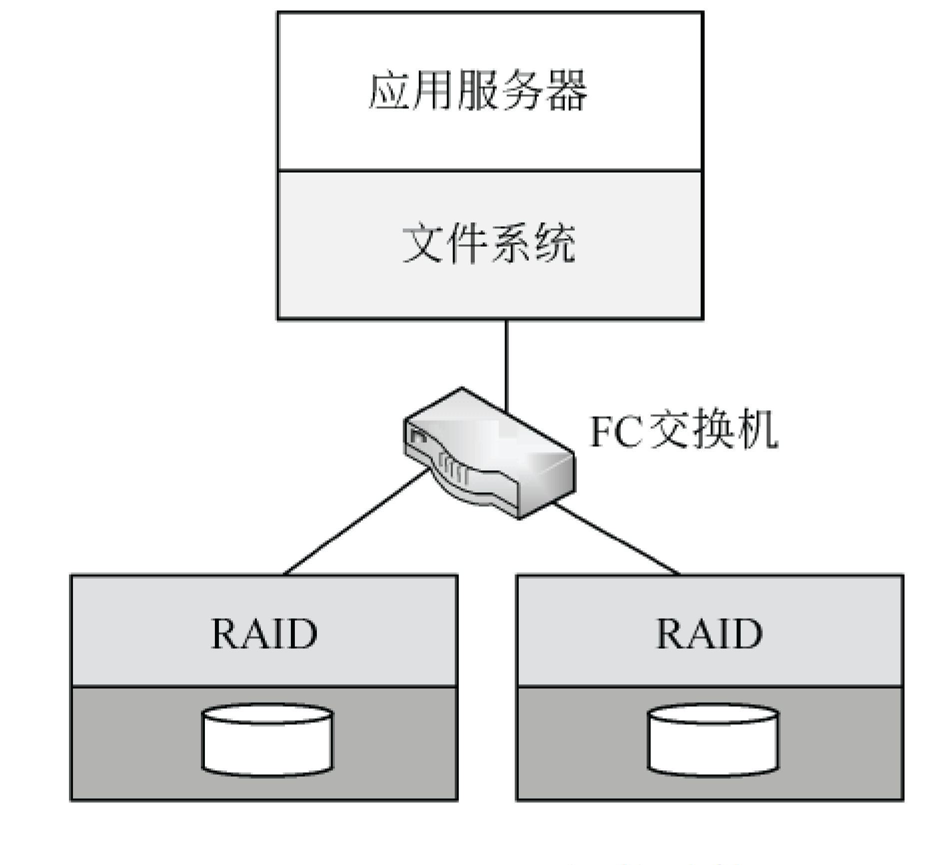


图 9-75 SAN 拓扑结构

与 NAS 相比，SAN 具有下面的特点。

（1）SAN 具有无限的扩展能力。由于 SAN 采用了网络结构，文件服务器可以访问 SAN 网络上的任何一个存储设备，因此用户可以自由扩展磁盘阵列、磁带库和服务器等设备，使得整个系统的存储空间和处理能力可以按照用户需求不断扩大。

（2）SAN 采用了为大规模数据传输而专门设计的光纤通道技术，所以具有更高的传输速度和更快的处理能力。

图 9-76 表示的是用户存储文件的过程。当客户端把要存储的文件发送给文件服务器时，文件服务器不是把数据存储在本地的硬盘上，而是将其发送给 SAN 网络，如果光纤通道交换机存储在适当的存储设备上，这些文件可以自动地转发到其他存储设备上，以实现数据镜像和系统容灾。

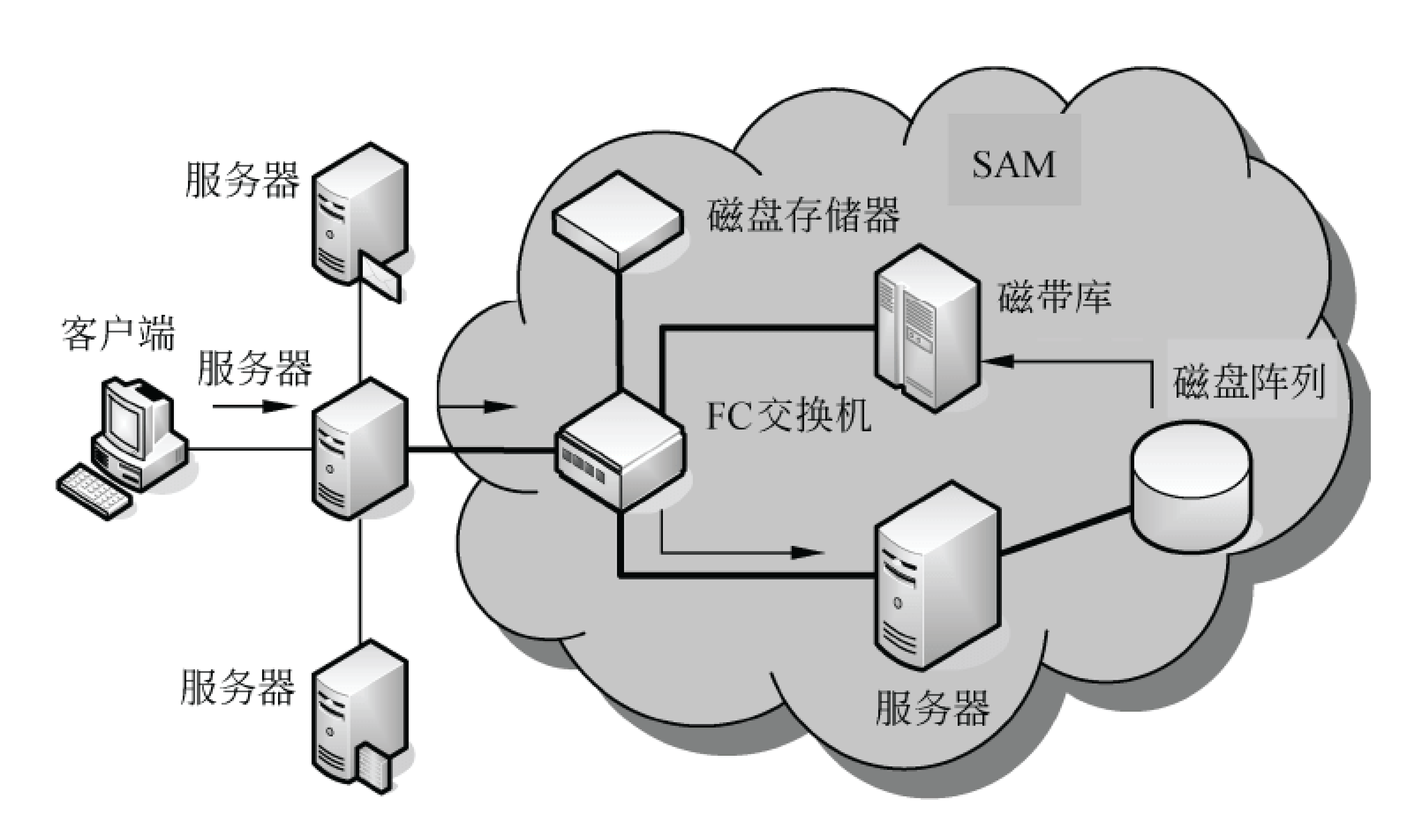
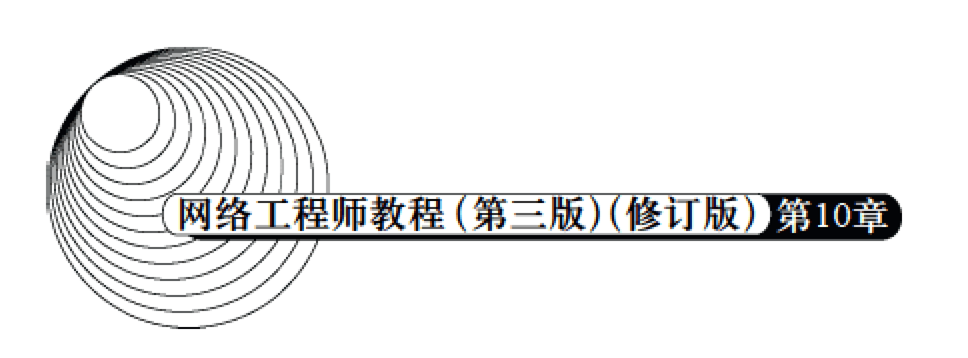


图 9-76 SAN 拓扑结构



第 10 章　网络规划和设计

网络规划和设计是根据网络建设的目标进行需求分析，设计网络的逻辑结构和物理结构，为网络工程的安装和配置准备各种技术文档。网络规划和设计过程是一个迭代和优化的过程，在网络的生命周期中这个过程重复多次，使得建成的网络能够适应技术的发展和应用的变化，为用户提供一个高效适用的网络计算平台。本章重点讲述网络分析和设计过程，并且介绍了结构化综合布线系统和网络故障诊断方法，最后给出了网络部署和配置的实例。

10.1　结构化布线系统

结构化综合布线系统（Structure Cabling System）是基于现代计算机技术的通信物理平台，集成了语音、数据、图像和视频的传输功能，消除了原有通信线路在传输介质上的差别。结构化综合布线系统包括建筑物综合布线系统（Premises Distribution System，PDS）、智能大厦布线系统（Intelligent Building System，IBS）和工业布线系统（Industry Distribution System，IDS）。这里要讲的是建筑物综合布线系统 PDS，这是一种能支持话音和数据通信，支持安全监控和传感器信号传输，支持多媒体和高速网络应用的电信系统，通过一次性布线提供各种通信线路，并且可以根据应用需求变化和技术发展趋势进行扩充，是一种技术先进、具有长远效益的解决方案。

结构化综合布线系统应满足下列要求。

- 标准化：采用国际、国家规范和标准来设计、施工和测试系统，采用符合国际和国家标准、得到国际权威机构认证的产品。
- 实用性：针对实际应用的需要和特点来建设系统，保证系统能满足现在和将来应用的需要。
- 先进性：采用国际最新技术，系统设计应具有一定的超前意识，保证在 5 至 10 年内技术上不落后。
- 开放性：充分考虑整个系统的开放性，系统要兼容不同类型的信号，适应各种网络拓扑结构和各种应用的要求。
- 结构化、层次化：易于管理和维护系统，应具有充足的扩展余地，具有一定的灵活性、较强的可靠性和容错性。

结构化布线系统分为 6 个子系统：工作区子系统、水平子系统、干线子系统、设备间子系

统、管理子系统和建筑群子系统，如图 10-1 所示。

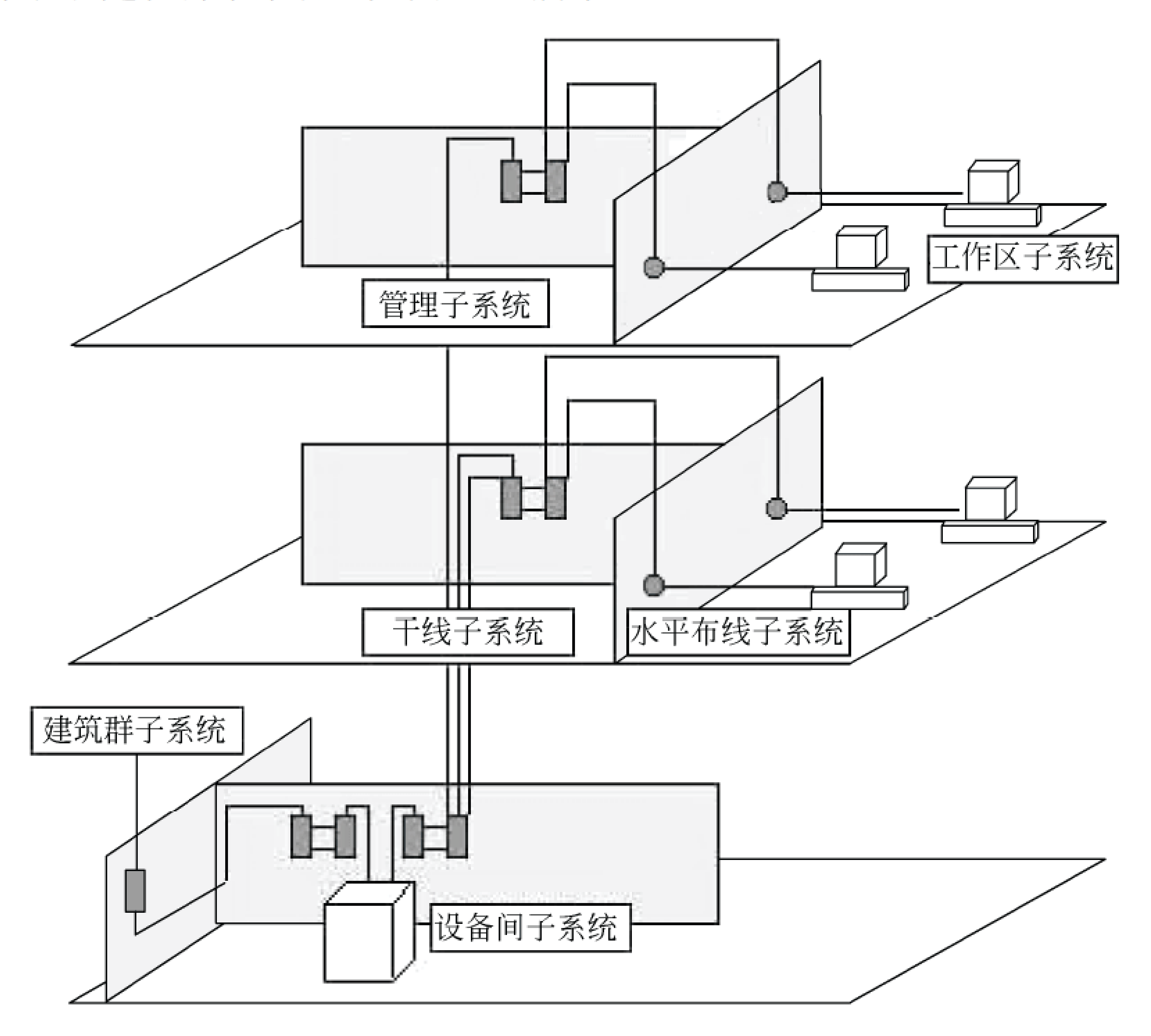


图 10-1　结构化布线示意图

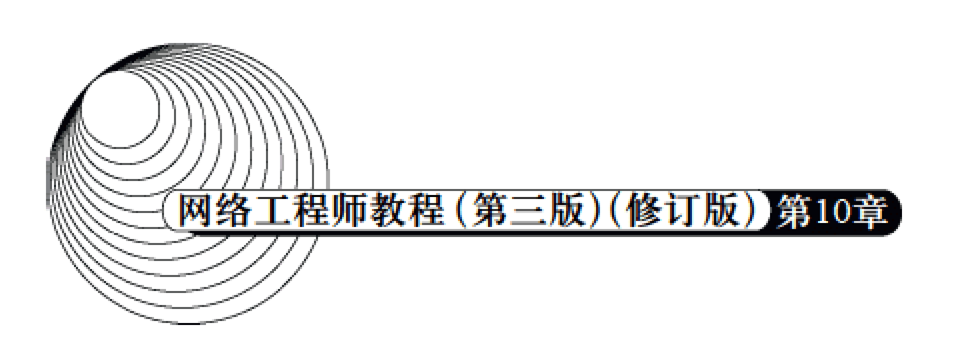
1．工作区子系统（Work Location）

工作区子系统是由终端设备到信息插座的整个区域。一个独立的需要安装终端设备的区域划分为一个工作区。工作区应支持电话、数据终端、计算机、电视机、监视器以及传感器等多种终端设备。

信息插座的类型应根据终端设备的种类而定。信息插座的安装分为嵌入式（新建筑物）和表面安装（老建筑物）两种方式，信息插座通常安装在工作间四周的墙壁下方，距离地面 30cm，也有的安装在用户办公桌上。通常一个信息插座需要 $9m^2$ 的空间。

2．水平子系统（Horizontal）

各个楼层接线间的配线架到工作区信息插座之间所安装的线缆属于水平子系统。水平子系统的作用是将干线子系统线路延伸到用户工作区。在进行水平布线时，传输介质中间不宜有转折点，两端应直接从配线架连接到工作区的信息插座。水平布线的布线通道有两种：一种是暗



管预埋、墙面引线方式，另一种是地下管槽、地面引线方式。前者适用于多数建筑系统，一旦铺设完成，不易更改和维护；后者适合于少墙多柱的环境，更改和维护方便。

3．管理子系统（Administration）

管理子系统设置在楼层的接线间内，由各种交连设备（双绞线跳线架、光纤跳线架）以及集线器和交换机等交换设备组成，交连方式取决于网络拓扑结构和工作区设备的要求。交连设备通过水平布线子系统连接到各个工作区的信息插座，集线器或交换机与交连设备之间通过短线缆互连，这些短线被称为跳线。通过跳线的调整，可以对工作区的信息插座和交换机端口之间进行连接切换。

高层大楼采用多点管理方式，每一楼层要有一个配线间，用于放置交换机、集线器以及配线架等设备。如果楼层较少，宜采用单点管理方式，管理点就设在大楼的设备间内。

4．干线子系统（Backbone）

干线子系统是建筑物的主干线缆，实现各楼层设备间子系统之间的互连。干线子系统通常由垂直的大对数铜缆或光缆组成，一头端接于设备间的主配线架上，另一头端接在楼层接线间的管理配线架上。

主干子系统在设计时，对于旧建筑物，主要采用楼层牵引管方式铺设，对于新建筑物，则利用建筑物的线井进行铺设。

5．设备间子系统（Equipment）

建筑物的设备间是网络管理人员值班的场所，设备间子系统由建筑物的进户线、交换设备、电话、计算机、适配器以及保安设施组成，实现中央主配线架与各种不同设备（如 PBX、网络设备和监控设备等）之间的连接。

在选择设备间的位置时，要考虑连接方便性，要考虑安装与维护的方便，设备间通常选择在建筑物的中间楼层。设备间要有防雷击、防过压过流的保护设备，通常还要配备不间断电源。

6．建筑群子系统（Campus）

建筑群子系统也叫园区子系统，它是连接各个建筑物的通信系统。大楼之间的布线方法有三种，一种是地下管道敷设方式，管道内敷设的铜缆或光缆应遵循电话管道和入孔的各种规定，安装时至少应预留 1 到 2 个备用管孔，以备扩充之用。第二种是直埋法，要在同一个沟内埋入通信和监控电缆，并应设立明显的地面标志。最后一种是架空明线，这种方法需要经常维护。

在进行结构化布线系统设计时，要注意线缆长度的限制，表 10-1 是 EIA/TIA-568 标准提出的布线距离最大值。

表 10-1　布线距离

子　系　统	光纤（m）	屏蔽双绞线（m）	无屏蔽双绞线（m）
建筑群（楼栋间）	2000	800	700
主干（设备间到配线间）	2000	800	700
配线间到工作区信息插座		90	90
信息插座到网卡		10	10

10.2　网络分析与设计过程

10.2.1　网络系统生命周期

一个网络系统从构思开始，到最后被淘汰的过程称为网络生命周期。一般来说，网络生命周期至少应包括网络系统的构思和计划、分析和设计、运行和维护的过程。网络系统的生命周期与软件工程中的软件生命周期非常类似，首先它是一个循环迭代的过程，每次循环迭代的动力都来自于网络应用需求的变更。其次，每次循环过程中都存在需求分析、规划设计、实施调试和运营维护等多个阶段。有些网络仅仅经过一个周期就被淘汰，而有些网络在存活过程中经过多次循环周期，一般来说，网络规模越大、投资越多，则其可能经历的循环周期也越长。

每一个迭代周期都是网络重构的过程，不同的网络设计方法，对迭代周期的划分方式是不同的，拥有不同的网络文档模板，但是实施后的效果都满足了用户的网络需求。常见的迭代周期构成方式主要有如下三种。

1．四阶段周期

四阶段周期能够快速适应新的需求变化，强调网络建设周期中的宏观管理，4 个阶段的划分如图 10-2 所示。

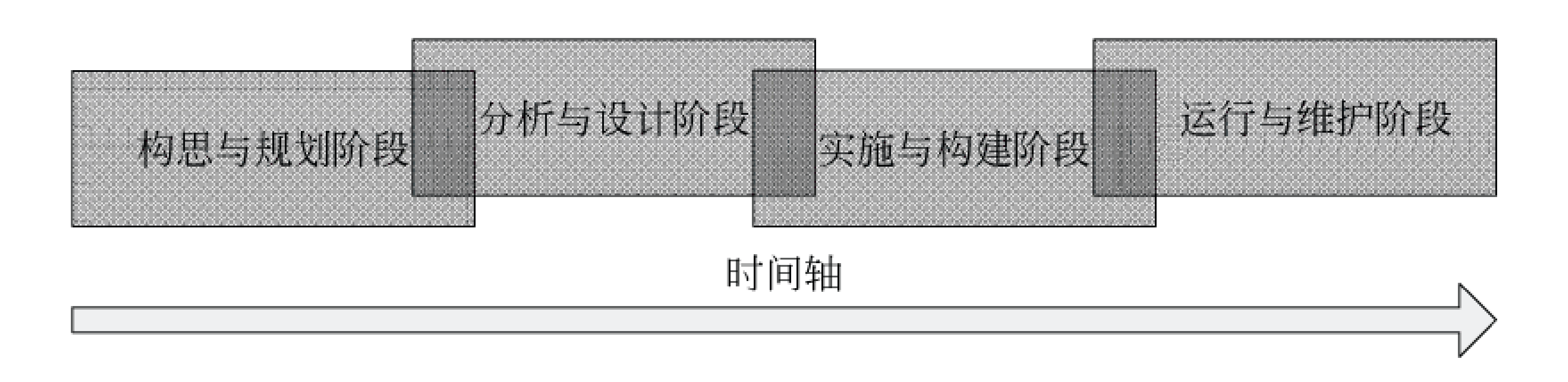


图 10-2　四阶段周期

4 个阶段分别为构思与规划阶段、分析与设计阶段、实施与构建阶段和运行与维护阶段，

这 4 个阶段之间有一定的重叠，保证了两个阶段之间的交接工作，同时也赋予了网络工程设计的灵活性。

构思与规划阶段的主要工作是明确网络设计的需求，同时确定新网络的建设目标。分析与设计阶段的工作在于根据网络的需求进行设计，并形成特定的设计方案。实施与构建阶段的工作在于根据设计方案进行设备购置、安装、调试，建成可试用的网络环境。运行维护阶段提供网络服务，并实施网络管理。

四阶段周期的长处在于工作成本较低、灵活性高，适用于网络规模较小、需求较为明确、网络结构简单的网络工程。

2．五阶段周期

五阶段周期是较为常见的迭代周期划分方式，将一次迭代划分为 5 个阶段。

（1）需求规范。

（2）通信规范。

（3）逻辑网络设计。

（4）物理网络设计。

（5）实施阶段。

在 5 个阶段中，由于每个阶段都是一个工作环节，每个环节完毕后才能进入到下一个环节，类似于软件工程中的“瀑布模型”，形成了特定的工作流程。如图 10-3 所示。

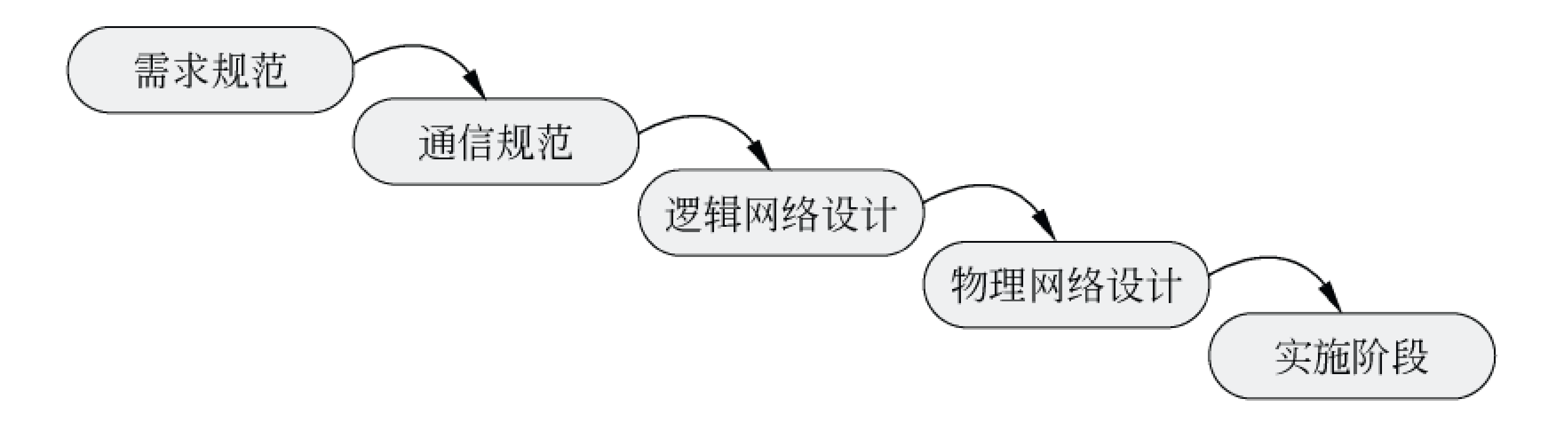


图 10-3　五阶段周期

按照这种流程构建网络，在下一个阶段开始之前，前一阶段的工作已经完成，一般情况下，不允许返回到前面的阶段，如果出现前一阶段的工作没有完成就开始进入下一个阶段，则会对后续的工作造成较大的影响，甚至引起工期拖后和成本超支。

这种方法的主要优势在于所有的计划在较早的阶段完成，系统的所有负责人对系统的具体情况以及工作进度都非常清楚，更容易协调工作。

五阶段周期的缺点是比较死板，不灵活。因为往往在项目完成之前，用户的需求经常会发生变化，这使得已开发的部分需要经常修改，从而影响工作的进程。所以基于这种流程完成网

络设计时，用户的需求确认工作非常重要。

五阶段周期由于存在较为严格的需求和通信分析规范，并且在设计过程中充分考虑了网络的逻辑特性和物理特性，因此较为严谨，适用于网络规模较大、需求较为明确、需求变更较小的网络工程。

3．六阶段周期

六阶段周期是对五阶段周期的补充，是对其缺乏灵活性的改进，通过在实施阶段前后增加相应的测试和优化过程，来提高网络建设工程中对需求变更的适应性。

6 个阶段分别由需求分析、逻辑设计、物理设计、设计优化、实施及测试、监测及性能优化组成，如图 10-4 所示。

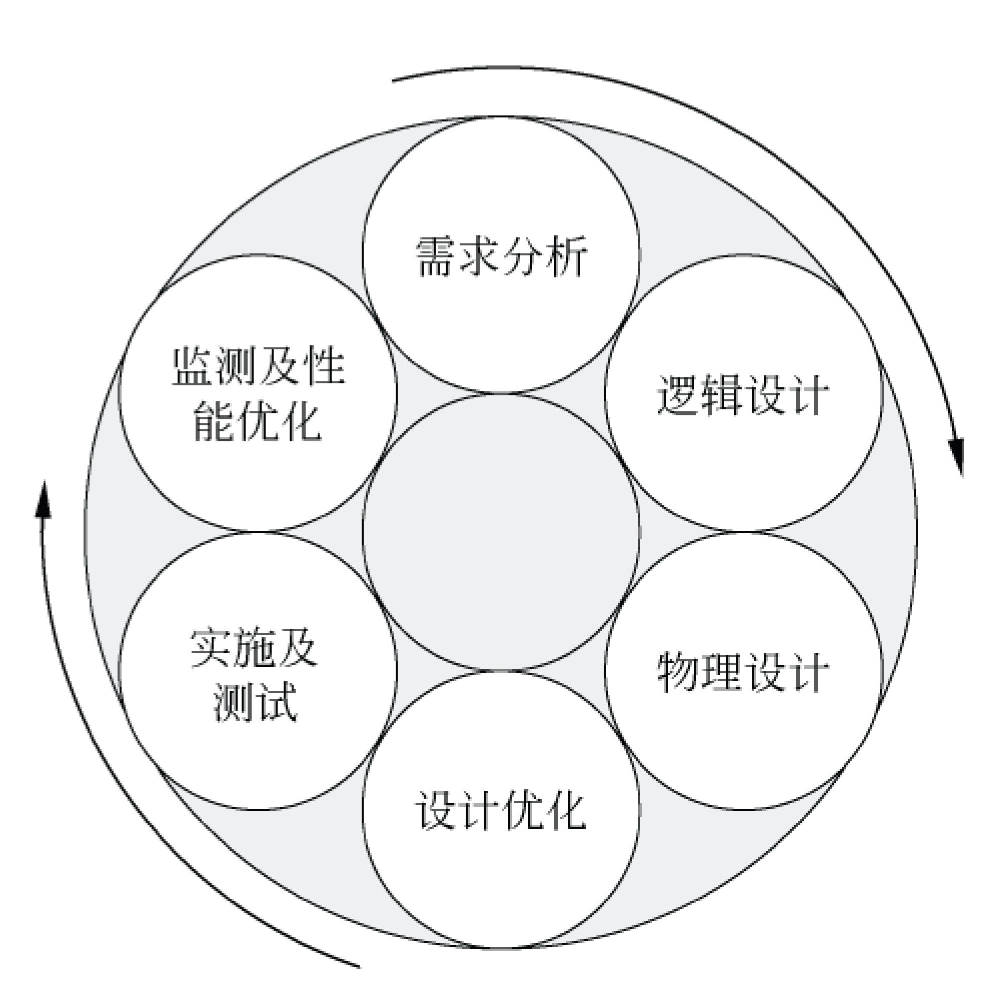


图 10-4　六阶段周期

在需求分析阶段，网络分析人员通过与用户进行交流来确定新系统（或升级系统）的商业目标和技术目标，然后归纳出当前网络的特征，分析当前和将来的网络通信量、网络性能、协议行为和服务质量要求。

逻辑设计阶段主要完成网络的拓扑结构、网络地址分配、设备命名规则、交换及路由协议选择、安全规划、网络管理等设计工作，并且根据这些设计选择设备和服务供应商。

物理设计阶段是根据逻辑设计的结果选择具体的技术和产品，使得逻辑设计成果符合工程设计规范的要求。

设计优化阶段完成工程实施前的方案优化，通过召开专家研讨会、搭建试验平台、网络仿真等多种形式找出设计方案中的缺陷，并进一步优化。

实施及测试阶段根据优化后的方案购置设备、进行安装、调试与测试工作，通过测试和试用后发现网络环境与设计方案的偏差，纠正其中的错误，并修改网络设计方案。

监测及性能优化阶段是网络的运营和维护阶段。通过网络管理、安全管理等技术手段，对网络是否正常运行进行实时监控，如果发现问题，则通过优化网络设备配置参数来达到优化网络性能的目的。如果发现网络性能无法满足用户的需求，则进入下一迭代周期。

六阶段周期偏重于网络的测试和优化，侧重于网络需求的不断变更，由于其严格的逻辑设计和物理设计规范，使得这种模式适合于大型网络的建设工作。

10.2.2　网络开发过程

网络开发过程描述了开发网络时必须完成的基本任务，而网络生命周期为描绘网络项目的开发提供了特定的理论模型，因此网络开发过程是指一次迭代过程。

由于一个网络工程项目从构思到最终退出应用，一般会遵循迭代模型，经历多个迭代周期。每个周期的各种工作可根据新网络的规模采用不同的迭代周期模型。例如在网络建设初期，由于网络规模比较小，因此第一次迭代周期的开发工作应采用四阶段模式。随着应用的发展，需要基于初期建成的网络进行全面的网络升级，则可以在第二次迭代周期中采用五阶段或六阶段的模式。

由于中等规模的网络较多，并且应用范围较广，因此主要介绍五阶段迭代周期模型。这种模型也部分适用于要求比较单纯的大型网络，而且采用六阶段周期时也必须完成五阶段周期中要求的各项工作。

将大型问题分解为多个小型可解的简单问题，这是解决复杂问题的常用方法。根据五阶段迭代周期的模型，网络开发过程可以被划分为如下 5 个阶段。

（1）需求分析。

（2）现有的网络体系分析，即通信规范分析。

（3）确定网络逻辑结构，即逻辑网络设计。

（4）确定网络物理结构，即物理网络设计。

（5）安装和维护。

因此，网络工程被分解成为多个容易理解、容易处理的部分，每个部分的工作构成一个阶段，各个阶段的工作成果都将直接影响到下一阶段的工作开展，这就是五阶段周期被称为流水线的真正含义。

在这 5 个阶段中，每个阶段都必须依据上一阶段的成果完成本阶段的工作，并形成本阶段的工作成果，作为下一阶段的工作依据。这些阶段成果分别为需求规范、通信规范、逻辑网络设计和物理网络设计文档。在大多数网络工程中，网络开发过程可以用图 10-5 来描述。

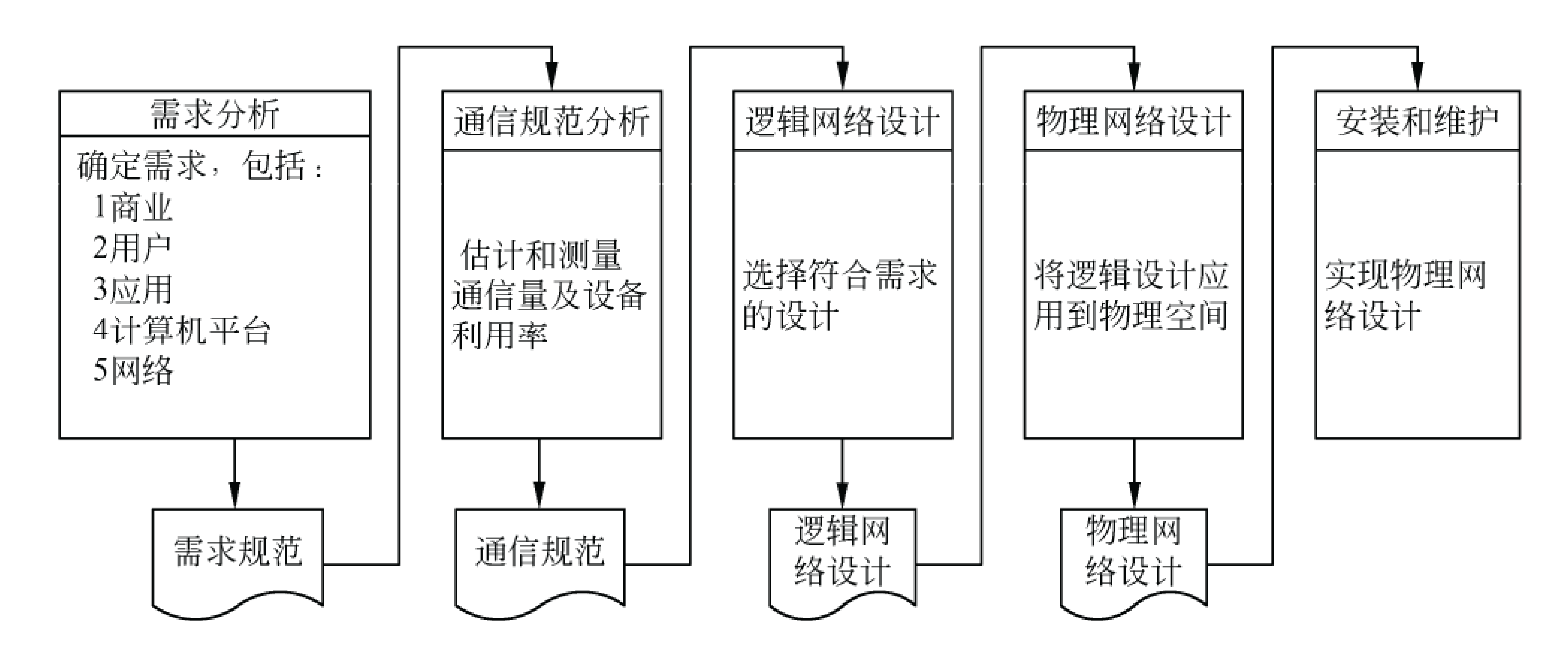


图 10-5　五阶段网络开发过程

下面详细介绍网络开发过程的各个阶段，只有理解了开发网络项目的各个阶段，才可以在实际开发过程中灵活运用。

1．需求分析

需求分析是开发过程中最关键的阶段，所有工程设计人员都清楚，如果在需求分析阶段没有明确需求，则会导致以后各阶段的工作严重受阻。需求阶段需要克服需求收集的困难，很多时候用户不清楚具体需求是什么，或者需求渐渐增加而且经常发生变化，需求调研人员必须采用多种方式与用户交流，才能挖掘出网络工程的全面需求。

收集需求信息要和不同的用户（包括经理人员和网络管理员）进行交流，要把交流所得信息进行归纳解释，去伪存真。在这个过程中，很容易出现不同用户群体之间的需求是矛盾的，特别是网络用户和网络管理员之间会出现分歧。网络用户总是希望能够更多、更方便地享用网络资源，而网络管理员则更希望网络稳定和易于管理。网络设计人员要在设计工作中根据工程经验，均衡考虑各方利益，才能保证最终的网络是可用的。

收集需求信息是一项费时的工作，也不可能很快产生非常明确的需求，但是可以明确需求变化的范围，通过网络设计的伸缩性保证网络工程满足用户的需求变化。需求分析有助于设计者更好地理解网络应该具有什么样的功能和性能，最终设计出符合用户需求的网络。

不同的用户有不同的网络需求，收集的需求范围如下。

（1）业务需求。

（2）用户需求。

（3）应用需求。

（4）计算机平台需求。

（5）网络通信需求。

（3）可以在路由器 Router1 上配置策略路由，其配置方法如下：

```
Router1(config)# access-list 1 permit 192.168.3.0 0.0.0.255
Router1(config)# access-list 2 permit 192.168.2.0 0.0.0.255
Router1(config)# route-map ISP1 permit 10
Router1(config-route-map)# match ip address 1
Router1(config-route-map)# set interface serial 0
Router1(config-route-map)# exit
Router1(config)# route-map ISP2 permit 20
Router1(config-route-map)# match ip address 2
Router1(config-route-map)# set interface serial 1
```

然后将每个路由图应用到路由器 Router1 的适当接口上，这里的适当接口是指那些数据流进入路由器的接口。

```
Router1(config)# interface fa0/0
Router1(config-if)# ip policy route-map ISP1
Router1(config-if)# interface fa0/1
Router1(config-if)# ip policy route-map ISP2
Router1(config-if)# interface fa0/2
Router1(config-if)# ip policy route-map ISP2
```

（4）可以在两个自治系统的边界路由器 Router1 上设置路由重发布，配置过程如下。

配置 OSPF 协议和路由重发布命令：

```
Router1(config)# router ospf 101
Router1(config-router)# redistribute rip subnets
Router1(config-router)# network X.X.X.X wildcard area 0
```

配置 RIP 协议和路由重发布：

```
Router1(config)# router rip
Router1(config-router)# network X.X.X.X //配置多条 network 命令
Router1(config-router)# passive-interface fa0/3
Router1(config-router)# redistribute ospf 101 match internal external 1 external 2
Router1(config-router)# default-metric 10
```